

GENESIS, un environnement pour la création musicale à l'aide de modèles physiques particulaires

Nicolas Castagné

▶ **To cite this version:**

Nicolas Castagné. GENESIS, un environnement pour la création musicale à l'aide de modèles physiques particulaires. Modélisation et simulation. Institut national polytechnique de Grenoble, 2002. Français. ⟨NNT : ⟩. ⟨tel-01274711⟩

HAL Id: tel-01274711

https://hal.science/tel-01274711v1

Submitted on 16 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque
|__/__/__/__/__/__/__/__/__/__/

T H E S E

pour obtenir le grade de

DOCTEUR DE L'Institut National Polytechnique de Grenoble

Spécialité:

« Acoustique, Traitement du signal et Informatique Appliqués à la Musique »

préparée au laboratoire Informatique et Création Artistique
dans le cadre de **l'Ecole Doctorale**
« Mathématiques, Sciences et Technologie de l'information, Informatique »
présentée et soutenue publiquement

par
Nicolas CASTAGNE

Le 07 octobre 2002

GENESIS
un environnement pour la création musicale à l'aide de modèles physiques particulaires

———

Directeur de thèse :
Claude CADOZ

———

JURY

M. Jean-Claude Risset	Rapporteur
M. René Caussé	Rapporteur
M. Claude Cadoz	Directeur de thèse
M. Michel Beaudouin-Lafon	Examinateur
M. Jean-Luc Schwartz	Examinateur
M. Arnaud Petit	Examinateur

J’exprime ma plus vive reconnaissance à Claude CADOZ, directeur de ma thèse. Son exigence scientifique, sa précision méthodologique et plus généralement ses réflexions rigoureuses et circonstanciées quant aux outils de création musicale, toutes choses qu’il a su me faire partager, sont pour beaucoup dans mes recherches et dans cet ouvrage. Je le remercie également de son encadrement attentif. Souvenirs très agréables des moments passés sur son invitation au Châtaignat.

J’ajoute Annie LUCIANI et Jean-Loup FLORENS à ces remerciements. Mes travaux s’inscrivent dans la continuité de ceux qu’ils ont entrepris aux cotés de Claude CADOZ, il y a quelques 25 années. Je le leur suis à tous trois redevable du cadre scientifique et technique et du contexte de pensée particulièrement riche et stimulant dans lequel cette thèse s’est déroulée. Merci à eux de m’avoir accueilli dans le laboratoire et d’avoir tour à tour participé avec enthousiasme et rigueur à mes réflexions.

Je tiens par ailleurs à saluer tous ceux, stagiaires ou permanents, qui ont participé au long de ces années à la conception et au développement de l’environnement GENESIS. Leur présence à fait du projet un travail d’équipe, avec tout le dynamisme qu’on peut en attendre, et donné lieu à de nombreux enrichissements mutuels. Sans leurs contributions fertiles, cette thèse n’aurait pu exister sous cette forme. Merci, donc, à Antoine CEZARD, Yannick CHAPUIS, Pirouz DJOHARIAN, Clément DURAND, Cyril GARINO, Florent LACHEROY, Cédric LEGRAND, Julien LOPEZ, Régis MONTE, Thomas PARET, Sylvain PATUREL, Peter TORVIK. Je salue tout particulièrement Peter et Pirouz pour l’importance de leurs contributions à mes réflexions.

Je remercie également vivement tous les compositeurs dont l’enthousiasme pour le projet a été une source permanente de motivation et d’idées, et parmi eux : Ludger BRUEMMER, Hans-Peter STUBBE, Giuseppe GAVAZZA, Mesias MAIGUASHCA, ainsi que les membres de la classe de composition du Conservatoire National de Région de Grenoble.

Je veux également saluer l’ensemble des artistes, étudiants, chercheurs, ingénieurs, administratifs et stagiaires qui ont animé l’ACROE durant ces années, et en particulier : Yannick CHAPUIS, Patrick FOURCADE, Olivier GIRAUD, Maria GUGLIELMI, Claire GUILBAUD, Nicolas TIXIER. Une pensée émue à Guy DIARD, et mes remerciements pour la calme efficacité dont il savait faire preuve dans toutes les situations difficiles.

J’exprime une vive gratitude à tous ceux qui ont donné du temps et de leurs compétences pour relire tout ou partie du manuscrit et offrir leurs conseils avisés : Pirouz DJOHARIAN, Patrick REBOUD, Peter TORVIK et mon père, René CASTAGNE.

J’adresse enfin mon affection toutes celles et ceux qui m’ont inlassablement encouragé dans l’accomplissement de cet ouvrage : mes proches, mes amis, ma famille.

Avant-Propos

Ce rapport présente le travail qui a conduit à la réalisation au sein du laboratoire ACROE-ICA de l'environnement informatique GENESIS pour la création musicale à l'aide du formalisme de simulation d'objets physiques CORDIS-ANIMA.

L'ensemble du projet relève du domaine 'Art – Science – Technologie' dans lequel s'élaborent des relations complexes entre des objectifs, des connaissances, des pratiques, des outils et des méthodes différentes mais dont l'articulation et la collaboration sont nécessaires. Dans le présent travail, c'est la musique, la physique, l'informatique et l'ordinateur qui sont principalement concernés.

Les rapports entre la musique, l'informatique et l'ordinateur ont déjà une histoire substantielle puisqu'ils ont débuté pratiquement avec l'apparition de ce dernier. Mais les relations sont multiformes, voire conflictuelles ou ambiguës. L'ordinateur doit-il être considéré comme un nouvel instrument lorsqu'il permet de synthétiser les sons ? Le compositeur doit-il s'effacer devant lui lorsqu'il 'compose automatiquement' ?

Avec l'ordinateur, de nombreux rapports établis de longue date ont été bouleversés. Dans l'approche qui nous anime ici, c'est au niveau du statut de l'ordinateur lui-même et dans l'observation du processus de création que nous reconsidérons les choses : l'ordinateur n'est envisagé ni comme un instrument ni comme un substitut de l'homme, mais comme un moyen de représentation, le plus général que nous ayons connu jusque là. Quant au processus de création, considérant qu'il est intimement lié et conditionné par les outils, tant matériels que conceptuels qui le supportent, nous l'envisageons fondamentalement comme un *dialogue*, comme une *interaction*. L'œuvre se façonne au contact de l'outil, de ses possibilités mais aussi des contraintes qu'il impose.

La conjonction de ces deux considérations détermine la perspective dans laquelle nous nous plaçons : dans la création musicale, le corps, l'instrument et le projet sont premiers, la théorie et les techniques, que ce soient celles de la notation, de l'écriture, de la composition leur donnent une extension pédagogique, sociale, culturelle, historique. Avec l'ordinateur, une *tabula rasa*, subie ou consentie, fut de toute façon nécessaire du simple fait qu'il fallut tout reprendre au niveau du *bit d'information*. En conséquence, il a été naturel de commencer par utiliser l'ordinateur pour *représenter l'instrument*. La suite, c'est-à-dire le principe de la simulation multisensorielle interactive des objets physiques qui est au cœur des travaux du laboratoire et de nos travaux, ainsi que tous les concepts et outils qui l'accompagnent, se déduisent naturellement de ce point de départ.

L'informatique musicale a été, depuis près d'un demi-siècle, le lieu d'apparition de nombreuses théories et de nombreux procédés, la source de nouvelles approches de la création musicale. Il nous semble aujourd'hui important de s'employer à porter les plus significatifs à leur stade de maturité. Le travail présenté ici, qui s'attache à un paradigme mis au jour il y a déjà vingt ans, voudrait se positionner résolument de cette façon : contribuer à porter le concept au stade de l'application, tant sur les plans artistique que pédagogique, que nous le croyons capable de soutenir.

Sommaire

Introduction ____ **9**

Partie I : Le paradigme de la simulation multisensorielle nteractive d'objets physiques ____ **15**

Chapitre 1 Le contexte de nos travaux ... 18
Chapitre 2 Nouveaux outils, nouvelles démarches ... 39
Conclusion : penser physique ... 59

Partie II : Interaction Homme-Machine et méthodologie de conception de GENESIS ____ **61**

Chapitre 3 Interface, Homme, Ordinateur ... 64
Chapitre 4 Guides et Principes ergonomiques ... 89
Chapitre 5 Méthode pour la réalisation de GENESIS ... 104
Conclusion ... 120

Partie III: Le processus de création avec GENESIS, Analyse des Tâches __ **123**

Chapitre 6 Un parcours simple pour la création ... 126
Chapitre 7 Considérations générales sur la création avec CORDIS-ANIMA ... 143
Chapitre 8 Phase de lutherie ; composition d'objets physiques ... 147
Chapitre 9 Phase de jeu et autres aspects de la création avec GENESIS ... 160
Résumé et Conclusion ... 165

Partie IV : Choix généraux pour l'environnement GENESIS ____ **167**

Chapitre 10 La version « GENESIS » de CORDIS-ANIMA ... 170
Chapitre 11 Organisation générale de GENESIS ... 201
Résume et Conclusion ... 220

Partie V : Fonctionnalités fondamentales de l'atelier de lutherie GENESIS ____ **223**

Chapitre 12 Représentation des objets CORDIS-ANIMA ... 226
Chapitre 13 Manipulation directe du réseau ... 244
Chapitre 14 Edition des paramètres et conditions initiales ... 256
Chapitre 15 Perspectives pour les fonctionnalités fondamentales ... 268
Résume et Conclusion ... 273

Partie VI : Fonctionnalités évoluées du mode Lutherie ____ **275**

Chapitre 16 Macro-modularité : Ensembles et capsules ... 278
Chapitre 17 Analyse modale et accordage ... 288
Chapitre 18 Outils complémentaires d'édition ... 303
Chapitre 19 Perspective pour les fonctionnalites evoluees ... 308
Résume et Conclusion ... 316

Partie VII : Simulation et phénomènes ____ **319**

Chapitre 20 La fenêtre de simulation ... 322
Chapitre 21 Perspectives pour les fonctionnalités de simulation ... 338

Conclusions et perspectives ____ **345**

Annexes ____ **353**

Références bibliographiques ____ **449**

Table des matières ____ **471**

INTRODUCTION

Ce travail s'insère dans le programme de recherche commun au laboratoire ICA et à l'association ACROE. L'Association pour la Recherche et la Création sur les Outils d'Expression a été créée en 1976 sous l'impulsion de Claude Cadoz, Annie Luciani et Jean-Loup Florens. Elle reçoit le soutien du Ministère de la Culture, en budget et en personnel. Le laboratoire Informatique et Création Artistique (INPG - UJF) a été créé en 1999. L'ACROE et l'ICA mènent en collaboration plusieurs missions de recherche, de développement, de valorisation, de formation et enfin de création.

La finalité d'ensemble du programme de recherche de l'ACROE-ICA est la création d'outils d'expression artistique dans les domaines de la musique, du mouvement et de l'image animée, et plus généralement la réalisation d'œuvres relevant des arts *instrumentaux du temps*. Une option fondamentale est le recours aux nouvelles technologies. Parmi elles, l'ordinateur occupe naturellement une place centrale.

Le laboratoire est le lieu d'une collaboration permanente entre les chercheurs et les créateurs (premiers bénéficiaires des recherches et utilisateurs privilégiés des outils). Celle-ci a donné naissance naturellement à une intense réflexion sur les processus de la création.

Ce programme de recherche recoupe plusieurs domaines scientifiques : l'informatique, la physique, l'automatique, l'électronique, l'électrotechnique, mais aussi la musicologie, la psychoacoustique, la synthèse d'image, l'animation, l'informatique musicale, etc.

Parmi ses thèmes de recherches, l'ACROE place la *relation instrumentale* en situation centrale. La relation instrumentale est celle que le musicien traditionnellement entretient avec l'instrument. Elle est de nature complexe, *multisensorielle* et *sensori-motrice*. C'est une *interaction* qui fait appel à l'acuité de plusieurs sens humains en collaboration et suppose une action motrice.

La *relation instrumentale* a été souvent écartée des utilisations musicales de l'ordinateur[1]. L'ACROE s'est donné comme objectif permanent de la réhabiliter, la plaçant ainsi en situation de référence.

L'ACROE-ICA a ainsi progressivement mis en place les éléments du paradigme de la *Simulation Multisensorielle et Interactive d'Objets Physiques*, dont les deux pièces maîtresses sont une famille de périphériques gestuels à retour d'effort (les Transducteurs Gestuels Rétroactifs, TGR©) et le formalisme CORDIS-ANIMA.

CORDIS-ANIMA est à la fois un langage, un système et un ensemble logiciel. Il permet de décrire des objets physiques par l'assemblage de modules élémentaires, puis de les simuler. Ces objets possèdent des propriétés mécaniques et dynamiques. Ils sont susceptibles d'interagir avec d'autres objets semblables ou, lorsque la simulation a lieu en temps-réel avec un opérateur humain via les périphériques.

[1] Une remarque similaire pourrait être faite pour d'autres arts instrumentaux abordés avec l'ordinateur.

Etant donné ce contexte et son potentiel de créativité, l'élaboration d'une interface utilisateurs spécifique s'avérait d'une nécessité vitale. C'est précisément à cet endroit que débutent nos travaux de recherche sur l'élaboration de GENESIS, un environnement pour la création musicale avec les objets physiques CORDIS-ANIMA.

Dans l'acception traditionnelle du terme, un environnement informatique accomplit deux tâches. Il est conçu pour orchestrer la mise en œuvre d'un ensemble d'algorithmes destinés à accomplir les étapes d'une fonction. Il réalise l'interface entre l'utilisateur et cette base logicielle dont il médiatise les concepts.

Un défi essentiel à relever dans l'élaboration de GENESIS vient précisément de la nouveauté des concepts qu'il s'agissait de médiatiser. La première étape de la mise en œuvre de CORDIS-ANIMA est en effet la création de l'objet physique. Cet assemblage et ses règles valent pour un nouveau langage d'écriture musicale dont les "objets" sont les nouveaux symboles. Ainsi ce paradigme va entraîner, de la part des utilisateurs – musiciens, interprètes ou créateurs – une démarche conceptuelle entièrement nouvelle autour de la notion de « pensée physique », démarche dont GENESIS a l'obligation d'être le support.

Au début de nos travaux, plusieurs environnements prototypes avaient déjà été développés en versions successives. Ils ont constitué un premier matériau pour alimenter notre réflexion.

Nos recherches se sont alors déroulées selon trois lignes de progression :

1) L'analyse de la nouveauté, c'est à dire la conduite d'une réflexion rigoureuse sur les voies par lesquelles le créateur mettant en œuvre la Simulation Multisensorielle Interactive d'Objets Physiques est amené à modifier profondément son processus de création.

2) La caractérisation précise et complète du processus de création musicale que devra soutenir GENESIS. Ceci a nécessité une analyse des tâches s'accomplissant lors de la création avec CORDIS-ANIMA.

3) enfin, la conception proprement dite du logiciel GENESIS.

Pour ce troisième axe nous aurons mené de front :

- une réflexion méthodique sur les fonctionnalités nécessaires
- des choix rigoureux, souvent restrictifs parmi ces fonctionnalités pour déterminer les plus importantes et ne retenir que celles qu'il était matériellement possible d'envisager dans un délai raisonnable.
- la définition précise de la manière dont ces fonctionnalités devront être rendues disponibles à l'utilisateur, c'est-à-dire sur leur ergonomie.
- l'implémentation effective sur machine et l'évaluation des résultats obtenus.

Une décision difficile mais essentielle et qu'il nous faut citer ici a été d'effectuer la plus grande part du développement de GENESIS autour du temps différé. Il est apparu que le repli sur le temps différé n'apportait aucune restriction au développement des fonctionnalités essentielles. Il permettait en revanche de mettre rapidement en ligne la version 1.5 de GENESIS actuellement disponible et utilisée. L'architecture de GENESIS a cependant été prévue pour faciliter la connexion au moteur temps réel, qui interviendra prochainement. Ce choix sera quoi qu'il en soit justifié en détail dans la suite du document.

Comme dans doute démarche de ce type, il nous aura également fallu identifier, rassembler puis mettre en œuvre un certain nombre de ressources.

Nous rangeons d'abord sous ce vocable les moyens issus du contexte expérimental du laboratoire, de ses acquis en recherche et des nombreux développements logiciels déjà effectués qu'il nous a fallu assimiler et parfois prolonger.

La seconde ressource viendra du dialogue constamment entretenu avec les utilisateurs du logiciel. Nous avons dans toute la mesure du possible tiré parti de remarques, de leurs souhaits et plus généralement d'une analyse de leurs travaux avec les versions successives de GENESIS.

Une recherche bibliographique constante nous aura permis de bénéficier d'une vision critique des nouveaux outils de l'informatique musicale au moment de choix décisifs pour GENESIS. Elle nous aura permis également d'acquérir une connaissance approfondie des principes, des méthodes et des techniques de l'Interaction Homme-Machine et de l'Ergonomie.

Enfin, pour la conception de GENESIS et sa mise en œuvre, nous avons eu recours à une démarche à la fois *itérative* en ce qu'elle alternait systématiquement les phases de développement et les phases d'évaluation du logiciel et *incrémentale* en ce qu'elle accède par étapes à la complexité finale.

L'organisation générale du document respecte en grande partie les étapes que nous venons de décrire.

Les deux premières parties fixent les objectifs et la démarche de notre travail. Ils le positionnent par rapport aux contextes historique, scientifique et technologique dans lesquels il s'est déroulé.

Une partie charnière est consacrée à l'analyse des tâches impliquées dans la mise en œuvre de GENESIS.

Les cinq dernières parties portent sur les choix effectués, la démarche de construction progressive et finalement la structure de GENESIS 1.5.

	Introduction	**9**
Contexte et bibliographie	**Partie I : Le paradigme de la simulation multisensorielle nteractive d'objets physiques**	**15**
	Partie II : Interaction Homme-Machine et méthodologie de conception de GENESIS	**61**
	Partie III: Le processus de création avec GENESIS, Analyse des Tâches	**123**
Conception de GENESIS	**Partie IV : Choix généraux pour l'environnement GENESIS**	**167**
	Partie V : Fonctionnalités fondamentales de l'atelier de lutherie GENESIS	**223**
	Partie VI : Fonctionnalités évoluées du mode Lutherie	**275**
	Partie VII : Simulation et phénomènes	**319**
	Conclusions et perspectives	**345**

Figure 1 : Organisation générale du document

Partie I

Le paradigme de la simulation multisensorielle interactive d'objets physiques

> [L'ordinateur] est un système ouvert et intégré, qui assume les interactions matériaux / organisations et mène à un nouveau terme provisoire l'artifice d'écriture dans la musique occidentale, au service d'une esthétique sans entrave, libre de construire ses propres conditions de possibilités, ses propres univers.
>
> [Barrière.90]

Cette première partie est consacrée au contexte technique et conceptuel de nos travaux.

Le premier chapitre introduit le paradigme de la Simulation Interactive et Multisensorielle d'Objets Physiques. Il précise les raisons qui ont amené l'ACROE à placer la relation instrumentale en situation de référence puis décrit dans ses lignes majeures le formalisme de modélisation physique CORDIS-ANIMA qu'interface GENESIS.

Le second chapitre situe la Simulation Interactive et Multisensorielle d'Objets Physiques en perspective avec les mutations récentes des outils de création sous l'impact des nouvelles technologies. Une analyse bibliographique nous conduit à dégager les caractères innovants et les apports fondamentaux de l'approche. Elle nous permet de comprendre en quoi la pratique de CORDIS-ANIMA nécessite une nouvelle démarche intellectuelle, que GENESIS devra accompagner.

Introduction

Contexte et bibliographie

Partie I : le paradigme de la simulation interactive et multisensorielle d'objets physiques

Chapitre 1 Le contexte de nos travaux 18

1.1 - LA RELATION INSTRUMENTALE COMME SITUATION DE REFERENCE 18
1.2 - PERIPHERIQUES GESTUELS - RETOUR D'EFFORT 20
1.3 - CORDIS-ANIMA 22
1.4 - ENVIRONNEMENTS POUR LA CREATION 34
1.5 - CONCLUSION: ICI APPARAIT LE PARADIGME DE LA SIMULATION MULTISENSORIELLE INTERACTIVE D'OBJET PHYSIQUE 38

Chapitre 2 Nouveaux outils, nouvelles démarches 39

2.1 - LES INSTRUMENTS ELECTROACOUSTIQUES 40
2.2 - L'ENREGISTREMENT SONORE 42
2.3 - LA SYNTHESE, L'ANALYSE, LA PSYCHOACOUSTIQUE 44
2.4 - LA COMPOSITION ASSISTEE PAR ORDINATEUR ET LES PROCESSUS D'ENGENDREMENT 46
2.5 - LES INTERFACES GESTUELLES, LES SYSTEMES TEMPS REEL 48
2.6 - LA SIMULATION INTERACTIVE ET MULTISENSORIELLE D'OBJETS PHYSIQUES 51

Conclusion : penser physique 59

Partie II : Interaction Homme-Machine

Partie III : Le processus de création avec GENESIS. Analyse des Tâches

Conception de GENESIS

Partie IV : Choix généraux pour l'environnement GENESIS

Partie V : Fonctionnalités fondamentales de l'atelier de lutherie GENESIS

Partie VI : Fonctionnalités évoluées du mode lutherie

Partie VII : Simulation et Phénomènes

Conclusions et perspectives

Chapitre 1

LE CONTEXTE DE NOS TRAVAUX

1.1 - La relation instrumentale comme situation de référence

En tant que phénomène communicationnel qui recourt à un mode d'émission complétant le mode vocal naturel, la musique fait nécessairement appel à ce qui est la fonction première de l'instrument : convertir une manifestation gestuelle en un phénomène sonore.

Dans la situation traditionnelle, si l'on entend par-là en absence de relais énergétique (amplification électrique), l'énergie du son émis prend sa source dans le geste de l'instrumentiste et nécessite une chaîne matérielle effectuant cette transformation. La matérialité de l'instrument est alors importante, car elle conditionne aussi bien les catégories de signaux sonores qui pourront être produits que la nature et l'allure des gestes qui les produiront. La relation instrumentale est donc au cœur du processus de communication en jeu (figure 1).

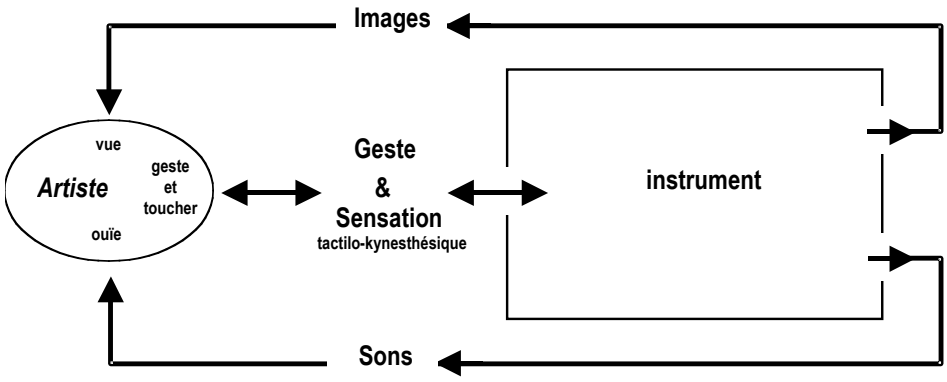


Figure 1 : la relation instrumentale

L'ACROE a très tôt postulé que la variété et l'expressivité des sons produits par synthèse nécessitaient la prise en compte de cette dimension instrumentale et matérielle[1] [Cadoz&al.81], [Cadoz.94b], [Cadoz.99b].

La *relation instrumentale* a été souvent tacitement ou volontairement écartée dans les utilisations musicales de l'ordinateur[2]. Tacitement parce que pendant longtemps, le temps réel qu'elle suppose n'était pas accessible. Volontairement parce que certains ont vu dans l'ordinateur la possibilité de s'affranchir du corps dans la création pour se consacrer aux aspects conceptuels et formels, et peut-être découvrir une nouvelle manière de faire la musique.

[1] Cette remarque vaut par ailleurs pour toutes les formes d'expression qui impliquent le corps face à un intermédiaire matériel, c'est-à-dire pour l'ensemble des arts instrumentaux du temps. Ce document concerne directement le son et la musique, mais les recherches menées au laboratoire concernent aussi les arts graphiques.

[2] Une remarque similaire peut à nouveau être faite pour d'autres arts instrumentaux abordés avec l'ordinateur.

Plus récemment, de nombreuses techniques et interfaces gestuelles conçues à des fins musicales ont été proposées pour tenter de reconstituer une relation instrumentale. Cependant, la correspondance qu'elles établissent entre geste et processus informatique reste souvent artificielle. Elles sont appliquées à des principes de synthèse ou de traitement du son qui n'intègrent en rien le principe de l'*interaction* gestuelle menant au concept de la relation instrumentale. Elles se résument le plus souvent au contrôle par gestes d'un processus qui auparavant était mené sans interaction. Elles opèrent généralement par l'ajout d'un *étage de contrôle temps-réel* externe au processus de synthèse lui-même.

A l'inverse, l'ACROE-ICA a considéré la Relation Instrumentale comme un principe fondateur de l'ensemble de ses recherches en informatique musicale. Dès 1976 elle est considérée comme une *situation de référence* maintes fois éprouvée, qu'il convient d'analyser et d'étudier. Sa réhabilitation devient un objectif central et générique.

L'ACROE-ICA a ainsi progressivement assemblé les éléments d'un outil informatique pour la création permettant de recréer les conditions d'une relation instrumentale. Essayons d'en définir rapidement les contours.

Par définition la relation instrumentale n'apparaît qu'en *phase de jeu*. Pendant cette phase l'ordinateur est nécessairement le lieu d'un calcul en temps réel ou *simulation.* Il doit à ce moment être capable d'interagir par *plusieurs* canaux sensoriels avec l'artiste : la simulation est donc devenue interactive et multisensorielle.

Cet outil va donc comprendre, outre le calculateur, un ensemble de périphériques matérialisant l'interaction multisensorielle. Le formalisme CORDIS-ANIMA intervient alors à deux niveaux. D'abord il va permettre de décrire l'instrument simulé. Il fournira ensuite l'algorithme de la simulation.

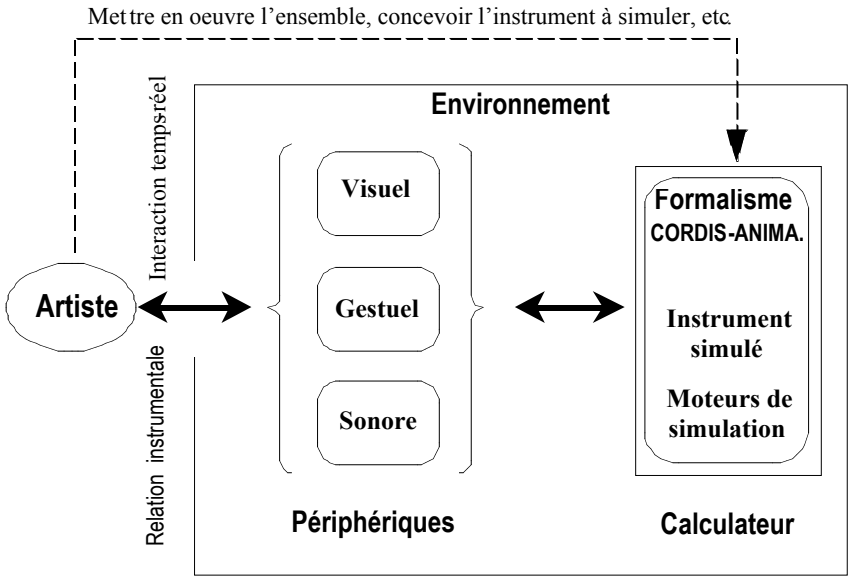


Figure 2 : les quatre éléments de l'outil de création
Calculateur, formalisme moteur de simulation, périphériques et environnement

Cependant, cet ensemble ne peut fonctionner que si un environnement en assure la cohérence et permet de pratiquer le langage CORDIS-ANIMA pour concevoir les instruments simulés.

Plusieurs environnements ont déjà été développés au laboratoire, suivant la finalité du système (son ou image) et le type d'utilisateur (chercheur ou artiste). C'est précisément sur le logiciel GENESIS, environnement utilisateur pour la version musicale de l'outil de création que nos recherches se sont portées.

Avant d'entrer dans la problématique conduisant à la conception de cet environnement nous décrirons rapidement les caractéristiques des deux éléments les plus importants qu'il devrait regrouper[1] : les périphériques (et parmi eux principalement le Clavier Rétroactif Modulaire TGR), et le formalisme CORDIS-ANIMA.

1.2 - Périphériques gestuels - retour d'effort

Par *périphériques*, nous entendons ici l'ensemble des dispositifs matériels qui, connectés au calculateur, permettent pendant la simulation en temps réel l'existence d'une relation instrumentale multisensorielle. L'intérêt du laboratoire pour le son et l'image l'a conduit à considérer les dispositifs communiquant avec trois sens humains : l'ouie, la vue et la perception tactilo-kinesthésique[2].

La nature et la technologie des périphériques dépendent en général des propriétés prises en considération dans la relation sensorielle.

Les périphériques visuels et sonores sont, pour l'essentiel, des périphériques courants : écran ou système de visualisation 2D, et système électroacoustique, le cas échéant additionnés d'une couche logicielle d'adaptation à la simulation.

Les recherches de l'ACROE ont ainsi principalement porté sur les périphériques gestuels où des avancées importantes ont été effectuées.

Le canal gestuel est fin, complexe. Il a fait l'objet d'une étude poussée, initiée dès la création du laboratoire [Gibet.87], [Cadoz.94b]. Elle a permis de dégager une typologie du geste instrumental qui a guidé la technologie des interfaces gestuelles. Nous en résumons les grandes lignes.

1.2.1 - Une Typologie du geste instrumental

L'observation de situations de jeu typiques montre que les gestes instrumentaux relèvent suivant leur fonction de trois catégories : sélection, modification et excitation [Cadoz.99b]. Il est intéressant de considérer ces trois catégories du point de vue de l'échange énergétique mis en jeu lors de la réalisation du geste.

Les gestes de sélection et de modification ne mettent en jeu, dans le cas général, qu'un très faible échange d'énergie. Ce sont en général des actions essentiellement porteuses d'information.

---

[1] Nous déciderons dans la suite de ne permettre dans les premières versions de GENESIS que des simulations en temps-différé. C'est le cas de la version 1.5 actuellement diffusée. Ce choix essentiel pour nos travaux sera justifié précisément. L'architecture de GENESIS a cependant été conçue pour faciliter la connexion au simulateur temps-réel, qui interviendra prochainement. Nous supposerons cette connexion réalisée dans les premières parties du document. Ainsi, nous caractériserons le potentiel de l'environnement GENESIS dans toute sa généralité, indépendamment des choix ultérieurs dictés par des considérations matérielles.

[2] Le sens tactilo-kinesthésique regroupe la perception tactile (sensibilité de la peau) et la perception des efforts (cellules nerveuses sensibles des muscles).

En revanche, le geste d'excitation se caractérise par un échange énergétique important, contrôlé par le musicien. Il entre dans la catégorie des gestes *ergotiques*.

Dans la situation traditionnelle c'est en effet par ce geste que s'effectue l'apport d'énergie que l'instrument transforme en phénomène acoustique. Mais l'essentiel ici est que cet apport d'énergie ne peut se faire que s'il y a réaction de la part de l'instrument[1]. Or c'est au travers de ce geste que l'instrumentiste va exprimer sa sensibilité. L'interaction cohérente qui existe entre l'instrument et l'instrumentiste engage une dépense d'énergie, un effort porteur d'expression. Ce dernier participe intimement à la genèse du son. Le canal gestuel n'est donc pas seulement un canal de communication. On doit le considérer également comme un canal d'interaction énergétique ou canal ergotique, maillon essentiel de la relation instrumentale.

1.2.2 - La famille des TGR©

Les périphériques gestuels simples et courants, tels la souris ou les périphériques MIDI, peuvent être utilisés pour la réalisation de gestes de sélection ou de modification mais ne sont pas suffisants.

Pour rendre compte d'un échange énergétique entre instrumentiste et objet virtuel, il faut pouvoir simuler la réaction de cet objet virtuel à l'effort de l'opérateur. Seul un système à *retour d'effort,* c'est à dire un ensemble capteur/effecteur dont la partie active est contrôlée par la simulation, permet de réaliser un tel couplage [Cadoz&al.84], [Cadoz&al.90b].

Les caractéristiques mécaniques et électromécaniques exigées par ces périphériques ont été déduites de l'étude parallèle de la sensibilité tactilo-kinesthésique *et* de la motricité humaine. Nous en résumons maintenant les principales.

La reproduction des transitoires est essentielle. Le canal doit donc offrir *une bande passante importante* au passage d'informations entre la simulation calculée et le périphérique gestuel, et ce dans les deux sens. Mais pour la même raison, le périphérique doit disposer d'une grande puissance en régime transitoire, afin de répondre correctement à des sollicitations brusques (percussion avec une extrémité dure, un ongle par exemple). On peut situer la bande passante en puissance de l'ensemble du système effecteur (électronique, électromécanique, mécanique) à plusieurs KHz, dans la mesure où, par exemple, les couches externes de l'épiderme humain sont sensibles dans cette plage de fréquence. La valeur pic de cette puissance mécanique doit être par ailleurs aussi proche que possible de la puissance musculaire humaine. Enfin, des contraintes relatives au système de captation du geste doivent en outre être prises en considération, comme l'encombrement, la modularité ou encore la possibilité d'une adaptation morphologique (touche, pantographe ou joystick).

---

[1] On peut noter que certains instruments nécessitent des intermédiaires entre le corps et la structure sonore : archet, plectre pour les gestes d'excitation, le système de pédale de la harpe de concert ou le slide de la guitare pour les gestes de modification… Ces intermédiaires gestuels font alors partie de l'instrument, et c'est avec eux qu'interagit l'artiste. Remarquons aussi que certains instruments ne requièrent qu'une ou deux des trois catégories de gestes. Les timbales, par exemple, ne nécessitent qu'un geste de d'excitation et de sélection du percuteur. Le cas de l'orgue d'église est particulier, puisqu'il ne demande pas de geste d'excitation (l'ouverture d'un tuyau étant la combinaison d'un geste de sélection et de modification). Dans le cas de l'orgue électronique ou du clavier-synthétiseur, l'expressivité du geste "d'excitation" reste très limitée, même si l'on a vu dernièrement se développer des touchers plus sensibles (vélocité ou 'after touch'). Une interaction gestuelle cohérente s'avère par contre particulièrement essentielle et porteuse d'expressivité dans le cas des gestes d'excitation entretenue, comme celui que demande le violon par exemple.

Face à ces contraintes très exigeantes, qui dépassent celles qui sont généralement retenue pour la conception des périphériques gestuels, l'ACROE-ICA développe une technologie spécifique qui prend corps dans famille de Transducteurs Gestuels Rétroactifs TGR©[1].

1.3 - CORDIS-ANIMA

Le formalisme CORDIS-ANIMA est le cadre formel de l'outil pour la création artistique. C'est à la fois un langage, c'est-à-dire un ensemble d'éléments et de règles de combinaison, un système de représentation, et un ensemble d'algorithmes. Il permet la conception, la représentation et la simulation des modèles sonores et visuels. Nous présentons ici les principes qui ont gouverné sa genèse puis nous résumons l'ensemble du formalisme [Cadoz&al.90a].

1.3.1 - Interface transducteur - simulation : point de communication

Le formalisme CORDIS-ANIMA porte l'empreinte des caractéristiques bi-directionnelles de la communication entre les périphériques, en particulier les TGR©, et la simulation.

Deux univers sont reliés par les périphériques : celui, réel, de l'instrumentiste, et celui de la simulation.

Pour pouvoir être reportée dans la simulation, l'interaction physique entre l'instrumentiste et le périphérique doit être traduite. En raison des caractéristiques du calculateur, elle ne peut l'être que sous la forme d'un nombre fini de signaux discrétisés.

Par ailleurs, la communication dans les systèmes numériques est par essence unidirectionnelle. Le caractère bidirectionnel de l'interaction entre la simulation et l'extérieur nécessite donc la circulation de deux ensembles de signaux, l'un *montant* l'autre *descendant*, transitant par deux canaux séparés et regroupés par un couple(s) d'entrée-sortie (figure 3).

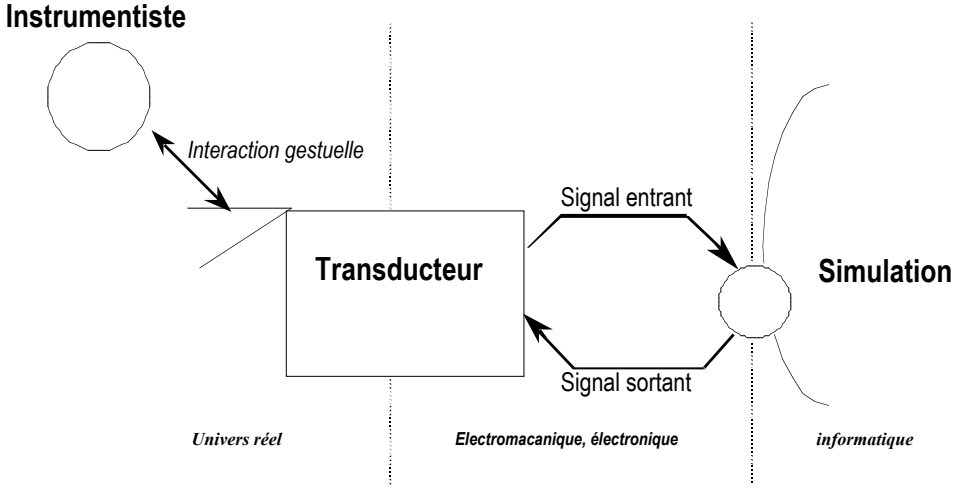


Figure 3 : L'interaction simulation/transducteur nécessite deux canaux unidirectionnels de sens opposés

[1] Transducteur Gestuel Rétroactif. Nom donné aux périphériques gestuels qui permettent de capter des mouvements gestuels pour les envoyer à l'ordinateur, et de produire des effets mécaniques (forces et/ou déplacements) commandés par l'ordinateur et destinés à la perception tactilo-proprio-kinesthésique de l'opérateur.

Le transducteur gestuel rétroactif (TGR©) est un capteur de *position* et un effecteur de *force* . Il communique donc avec la simulation selon ces deux dimensions mesurables. Comme des transducteurs capteur de force et effecteurs de position sont également envisageables, deux types de *points de communication* entre la simulation et le transducteur doivent en fait être considérés : celui fournissant un signal de position à la simulation et retournant un signal de force vers le périphérique, et son symétrique. Comme enfin ces points vont acquérir par la suite une grande importance, un nom leur a été attribué : respectivement point L et point M (figure 4).

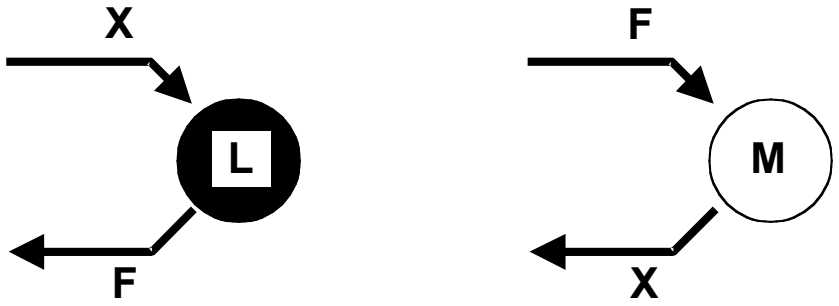


Figure 4 : Les deux types de point de communication entre la simulation et l'extérieur : point M et point L.
Les variables échangées sont des forces (F) et des positions (X)

1.3.2 - Principes générateurs du formalisme

Par ailleurs la conception du formalisme CORDIS-ANIMA répond à plusieurs principes :

- *Modularité* : le formalisme doit être modulaire. Il faut donc définir ce que sont les modèles minimaux ou *modules élémentaires*, qui constitueront les briques indivisibles.
- *Caractère générateur* : à l'inverse, ces modules élémentaires, associés à des règles d'assemblage, doivent pouvoir donner lieu à la génération de structures complexes.
- *Accès à l'expérimentation* : chaque module élémentaire doit être « expérimentable » au sens de la relation instrumentale. Son comportement doit donc être perceptible via les canaux gestuels, visuels et sonores. Cette propriété est naturellement extensible aux objets complexes.
- *Cohérence* : les principes de la communication simulation-transducteur sont étendus à la communication entre modules. Tout module doit donc être susceptible d'interagir avec son environnement (virtuel dans la simulation ou réel par le transducteur) via un canal montant et un canal descendant, porteurs respectivement de deux variables duales l'une étant nécessairement une force et l'autre position. Selon le même principe toute communication entre modules se fait via l'un des points de communication L ou M définis plus haut.
- *Economie* : les algorithmes des modules sont choisis de manière à présenter une complexité minimale.

1.3.3 - Réseaux d'interaction par les points M et L - règles de connexion

Des principes de cohérence et de modularité se déduit la forme générale *externe* d'un modèle CORDIS-ANIMA. Toute partie de modèle, qu'elle soit reliée à une autre partie de la simulation ou à un périphérique, communique avec l'extérieur via des points de communication de type M ou de type L (figure 5).

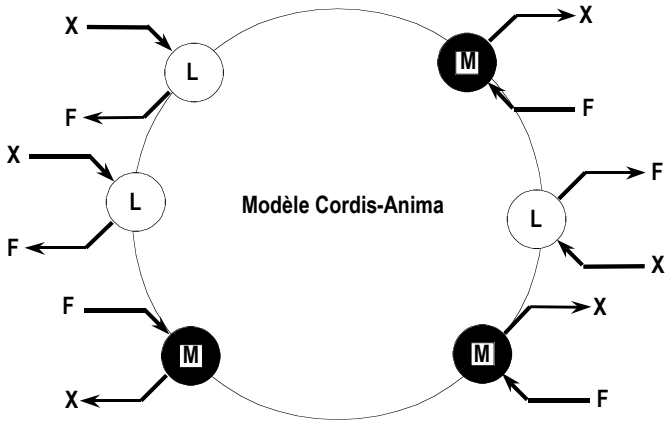


Figure 5 :* Surface *d'un modèle répondant au formalisme CORDIS-ANIMA

Le même principe de cohérence conditionne en outre les règles de connexion inter-modules via leurs points de communication. La force est une variable intensive : les sorties de force de plusieurs points L peuvent se sommer à l'entrée de force d'un point M. En revanche, il n'est pas possible de connecter les sorties de position de plusieurs points M à l'entrée d'un point L, car il n'est pas possible de donner un sens physique à la combinaison de plusieurs positions. Les règles de connexions s'énoncent alors de la manière suivante (figure 6) :

- La connexion de deux points de communication consiste à utiliser la sortie de l'un comme entrée de l'autre et réciproquement. En conséquence, ne sont connectables que des points de type opposés : un point M avec un point L.
- On peut connecter plusieurs points L à un même point M. La force entrante dans le point M est alors la somme des forces sortantes des divers points L. La position entrante dans les divers points L est la position sortante du seul point M commun auquel ils sont connectés.

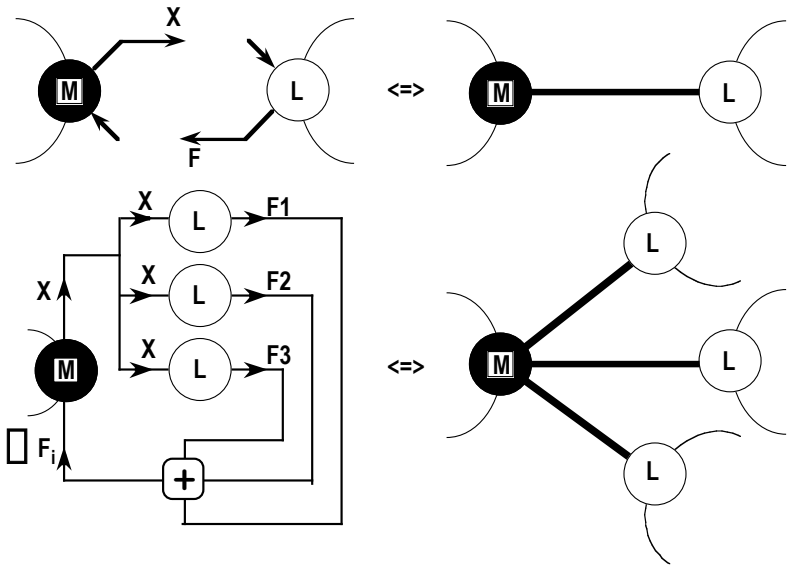


Figure 6 : Règles de connexion entre modèles
Sur ce schéma, les arcs de cercles représentent l'intérieur des modèles.
Un point de communication M peut être connecté à plusieurs points L.
Un point de communication L ne peut pas être connecté à plusieurs points M.

1.3.4 - Elément matériel <MAT> et élément de liaison <LIA>

La forme extérieure des modules élémentaires, c'est-à-dire leurs points de communication, se déduit alors du principe de modularité et de leur caractère générateur.

Si les modules élémentaires n'avaient qu'un point de communication, il serait impossible, compte tenu des règles de connexion, de générer des modèles complexes. Il faut donc introduire plusieurs points de communication dans un même module. Un rapide examen des cas possibles montre que l'unique base ayant le caractère générateur est constituée d'une part d'un module à un point M, d'autre part d'un module à deux points L.

Ces *types* de modules sont définis par leur « connectique », c'est à dire leur aptitude à échanger avec l'extérieur. Le premier, élaborant une position à partir d'une force, est appelé élément matériel ou <MAT>. Le second, qui permet de connecter entre eux les points matériels, est appelé élément de liaison ou <LIA>.

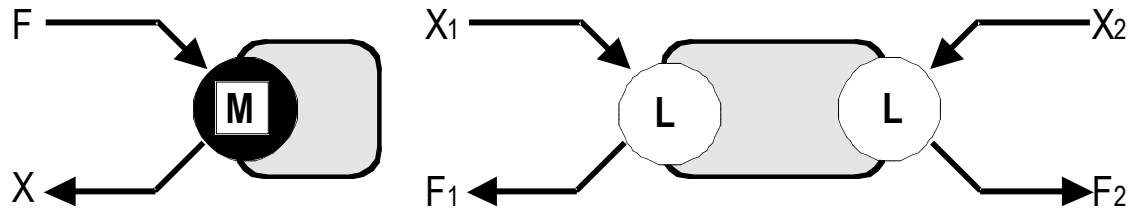


Figure 7 : forme externe des modules élémentaires <MAT> et <LIA>.
Du fait des règles de connexion, seule la combinaison de modules à 1 point M et de modules à deux points L possède le caractère générateur . La zone grise représente ici l'intérieur du module, et notamment son algorithme.

1.3.5 - Dimension de l'espace de simulation et spatialité des objets CORDIS-ANIMA

Jusqu'à maintenant, pour simplifier la présentation, les signaux de force et position échangés entre les modules par leur point de communication ont été supposés unidimensionnels.

En fait, CORDIS-ANIMA permet par construction de modéliser un espace de simulation tridimensionnel (voir n-dimensionnel avec n quelconque), dans lequel la position d'un point matériel est repérée sur trois axes (n axes) et les forces ont trois composantes (n composantes). Les signaux échangés entre les points de communication sont alors tridimensionnels (n-dimensionnels), et on parle de *vecteurs* force et *vecteurs* position.

Une simulation tridimensionnelle (dans laquelle les signaux sont tridimensionnels) communique avec le transducteur par trois paires de voies, chaque paire contenant une voie montante et une descendante. Chaque paire (montante - descendante) correspond sur le TGR© une touche de commande. Trois touches sont donc nécessaires pour interagir avec une simulation tridimensionnelle. Un joystick à trois degrés de liberté peut être monté sur ces trois touches et réaliser une adaptation morphologique. Il décomposera le mouvement de la main qui le contrôle selon trois composantes. Une matrice réversible de changement de repère géométrique permet de reconstituer, côté simulation, le mouvement de la main à partir des signaux des trois touches (figure 8).

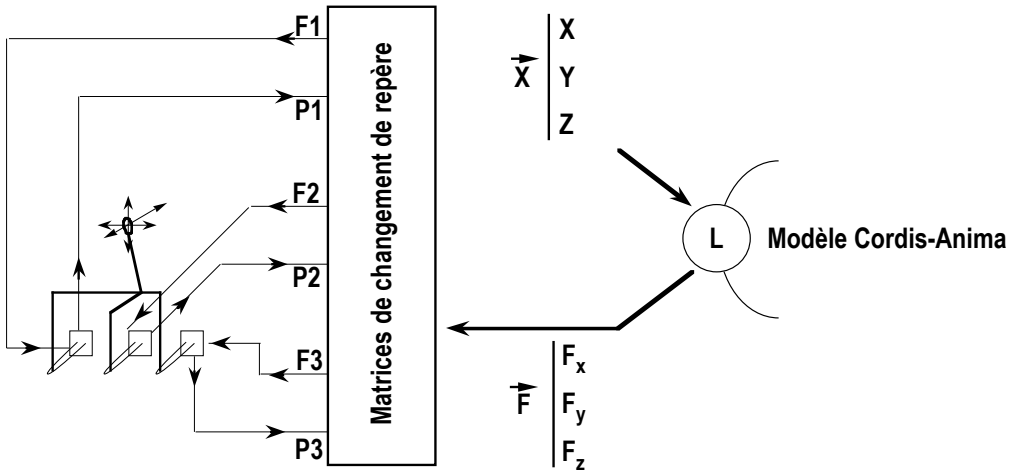


Figure 8 : communication TGR / simulation dans le cas d'une simulation 3D.
La matrice de changement de repère permet de transformer les coordonnées des trois touches dues au système d'adaptation morphologique en projection suivant les 3 axes de l'espace réel.

Les modules du formalisme CORDIS-ANIMA intègrent la notion de spatialité, en ce sens qu'ils peuvent être mono, bi ou tridimensionnels, c'est-à-dire échanger par leurs points de communication des signaux mono, bi ou tridimensionnels. L'utilisateur peut donc choisir la dimension de l'espace de simulation de ses modèles - nous dirons en raccourcis la dimension de ses modèles.

Lors de la mise en œuvre de CORDIS-ANIMA le choix de la dimension de l'espace de simulation revêt une certaine importance. Dans la suite, nous expliquerons pourquoi, nous avons décidé de fonder GENESIS, sur la version monodimensionnelle ou « *topologique* » de CORDIS-ANIMA[1].

1.3.6 - Principes généraux des algorithmes CORDIS-ANIMA

Nous résumons ici les principes mis en œuvre par les algorithmes de calcul des sorties en fonction des entrées dans chaque module CORDIS ANIMA.

T simulation

Du fait des propriétés des machines numériques, les signaux échangés entre des modules sont nécessairement des signaux échantillonnés. De plus, l'élaboration d'un échantillon de ces signaux nécessite un délai incompressible : entre un échantillon se présentant à l'entrée d'un module et l'échantillon qu'il élabore en sortie, un laps de temps s'écoule. Les algorithmes de CORDIS-ANIMA répondent alors aux caractéristiques suivantes, qui définissent la *T-simulation* (figure 9) :

- Un module calcule un signal synchrone échantillonné à la période T
- Entre l'entrée d'un signal et la sortie qui en résulte une seule période T s'écoule.

[1] Toutefois on pourra trouver dans l'annexe B le compte rendu de l'étude que nous avons effectuée, du potentiel des modèles spatiaux pour le son et la musique. Outre qu'elle ouvre la voie à des développements futurs, cette étude a permis d'introduire un nouveau type de liaison pour les modèles unidimensionnels, dont nous serons amenés à reparler.

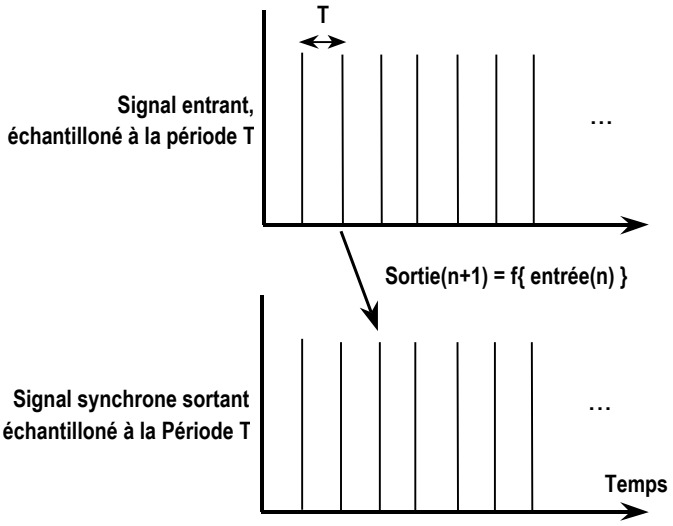


Figure 9 : T-simulation dans un module
l'échantillon en sortie à t=n est fonction de l'échantillon en entrée à t=n-1

Forme générale des algorithmes

Le principe d'économie impose alors qu'un module recevant en entrée un signal échantillonné E(n) et élaborant un signal S(n) (appelé *état* du module) utilise un minimum de valeurs du passé et un minimum d'opérateurs. Pour ces raisons, les règles de calcul de la sortie en fonction des entrées sont d'abord cherchées sous la forme linéaire :

$$S(n) = P1.S(n-1) + P2.S(n-2)+P3\ldots.. + Q1.E(n) + Q2.E(n-1)\ldots. + Cste$$

où P1, P2… Q1, Q2… et Cste sont les *paramètres* du module, des constantes dont on cherche autant que possible à minimiser le nombre.

Phase <MAT>, phase <LIA>

Au cours d'une période d'échantillonnage T (c'est-à-dire dans l'intervalle où se calcule l'état du modèle à t=n+1 à partir de l'état à t=n) deux phases se succèdent.

Dans la première, tous les modules de type <LIA> calculent les nouvelles forces en fonction des positions à t=n.

Dans la seconde, les algorithmes de tous les modules de type <MAT> sont exécutés et calculent, en fonction de ces nouvelles forces, les positions à t=n+1. La communication avec les transducteurs suit le même principe : pour le TGR, les forces à effectuer sont disponibles après la phase <LIA>, les positions mesurées pendant la phase <MAT>.

Une simulation CORDIS-ANIMA est donc une succession de phases <MAT> et <LIA>. Effectuer un pas de la simulation, c'est effectuer une phase <LIA> puis une phase <MAT>.

Action / réaction, interaction

Les points matériels <MAT> sont en interaction mutuelle via les modules de liaison. En se référant au principe d'action / réaction de la dynamique newtonienne, les règles de calcul des éléments de liaison <LIA> doivent produire, en sortie, *deux forces d'intensité et de direction égales mais de sens opposé*, dirigées respectivement vers chacun des <MAT> connectés.

1.3.7 - Principaux modules

Ces diverses propriétés étant exposées, nous pouvons décrire les modules élémentaires de CORDIS-ANIMA.

Les trois premiers reproduisent les comportements courants de la matière réelle : l'Inertie, l'Elasticité et la Viscosité. Viennent ensuite les modules d'interaction non-linéaires permettant de modéliser des comportements plus complexes et plusieurs modules fonctionnels.

Dans la suite, les signaux (X_n) et (F_n) peuvent être multidimensionnels – suivant la dimension de l'espace de simulation utilisé.

Modules physiques élémentaires

- **L'inertie : Module MAS**

Le module MAS, modélise le comportement inertiel de la matière.

Il s'agit d'un point matériel qui élabore sa position à partir de la force qui lui est appliquée. Il doit permettre d'expérimenter l'inertie à travers le TGR. La relation fondamentale de la dynamique projetée sur la forme générale des algorithmes C.A. donne :

Algorithme	$X_{n+1} = 2 * X_n - X_{n-1} + \frac{1}{M} F_n$
Paramètre	inertie *M*

On remarque que le carré de la fréquence d'échantillonnage n'apparaît pas explicitement dans cette équation. En fait, la fréquence d'échantillonnage étant constante, elle peut être reportée dans le paramètre M sans perte de généralité[1].

- **Le point fixe : module SOL**

Le module SOL est un module de type <MAT> dans lequel le paramètre d'inertie M serait infini : c'est un point matériel immobile quelles que soient les forces qu'il reçoit. Son algorithme est :

Algorithme	$X_{n+1}=X_n=X_0$
Paramètre	Néant (X_0 est une condition initiale)

[1] En d'autres termes, puisque le but n'est pas de *mesurer* mais de *rendre perceptible* une accélération, le terme T^2 qui apparaît lors de la discrétisation de la relation fondamentale de la dynamique est reporté dans le paramètre M. La base des paramètres CORDIS n'est pas identique à celle du physicien. Pour autant, il peut être utile de pouvoir considérer d'autres bases paramétriques, permettant par exemple de faire le lien avec des valeurs qui seraient *mesurées* sur des objets réels - ou encore avec les valeurs d'inerties mesurables via le TGR. Le chapitre 10 (partie IV) fait le point sur les diverses bases coexistant autour de CORDIS-ANIMA.

- **L'élasticité : module RES**

Le RES (RESsort) entre dans la catégorie des modules de type <LIA> . Il produit donc deux forces de sens opposés à partir des positions X1 et X2 des deux <MAT> auxquels il est connecté.

Pour modéliser le comportement élastique, l'algorithme suivant est utilisé :

Algorithme	$F_{n+1,2\to 1} = K*(d_n - l_0)*u_{1\to 2}$
Paramètres	raideur K, longueur au repos l_0
Avec :	$d_n = \left\| X_{n,1} - X_{n,2} \right\|$

- **La viscosité : module FRO**

Le FRO (FROttement) est <LIA> permettant d'expérimenter un comportement visqueux. Son algorithme vérifie :

Algorithme	$F_{n+1,2\to 1} = Z*(d_n - d_{n-1})*u_{1\to 2}$
Paramètre	viscosité Z
Avec là encore :	$d_n = \left\| X_{n,1} - X_{n,2} \right\|$

Modules physiques intégrés ; macro-modules

Pour des raisons pratiques, le formalisme définit deux macro-modules. Dans tous les cas, il est possible de les remplacer par une combinaison des modules précédents. Mais ils permettent une précieuse économie conceptuelle lors de la modélisation ainsi qu'en temps de calcul lors de la simulation. Le premier est la liaison visco-élastique REF, le second la cellule CEL topologique.

- **La liaison visco-élastique REF**

Le module REF (REssort-Frottement) est un <LIA> équivalent au regroupement d'un module RES et d'un module FRO connectés aux mêmes <MAT>. Son algorithme se déduit simplement des précédents :

Algorithme	$F_{n+1,2\to 1} = \left[K*(d_n - l_0) + Z*(d_n - d_{n-1})\right]*u_{1\to 2}$
Paramètres	raideur K, longueur au repos l_0 et viscosité Z

Le module REF est, de fait, le <LIA> le plus couramment employé dans les modèles CORDIS-ANIMA.

- **La cellule CEL topologique**

La cellule CEL n'est définie que lorsque l'espace de simulation est monodimensionnel et ne peut donc être utilisée que dans les modèles *topologiques*.

Dans un tel espace, elle est constituée d'un module SOL immobile à X=0, d'un module MAS et d'un module REF liant les deux précédents. Le module CEL ne présente à l'extérieur qu'un point de communication, de raideur M, qui est celui du module MAS qu'il contient (le SOL ne pouvant être connecté à un point L). L'algorithme consiste donc à calculer la position X(n) de ce point M en fonction des forces extérieures F(n) qu'il reçoit :

Algorithme	$X_{n+1} = 2 * X_n - X_{n-1} + \frac{K * X_n + Z * V_n + F_n}{M}$
Paramètres	inertie *M*, raideur *K* et viscosité *Z*.

La cellule CEL est donc *le plus petit modèle pouvant donner lieu à des oscillations amorties dans l'univers des modèles topologiques CORDIS-ANIMA*. Elle permet en outre d'exprimer simplement la présence d'une viscosité de milieu. C'est, de fait, le module de type <MAT> le plus courant dans les modèles topologiques.

Modules physiques non-linéaires

A ces modules linéaires sont ajoutés plusieurs modules non linéaires qui permettent, au sein du formalisme, de modéliser des comportements plus complexes.

- **Point de contact entre parties d'un modèle : module BUT**

Il faut pouvoir représenter l'entrée en contact de deux parties d'un modèle qui sont sans interaction dans le cas général. C'est le rôle du module BUT.

Le module BUT (BUTée) est une liaison viscoélastique qui n'est active que lorsque les deux <MAT> sont suffisamment proches pour être dans une zone définie d'interaction mutuelle. Hors de cette zone, l'interaction entre les deux <MAT> est nulle :

Algorithme

Si $d(X_{1,n}, X_{2,n}) > S$ *alors* $F_{n+1,2->1} = 0$

Si $d(X_{1,n}, X_{2,n}) <= S$ *alors*

$$F_{n+1,2->1} = \left[K * (d_n - S) + Z * (d_n - d_{n-1})\right] * u_{1->2}$$

Paramètres raideur *K*, viscosité *Z*, longueur au repos l_0, seuil *S*

Les utilisations de la liaison BUT sont multiples. Exprimant la mise en contact, elle permet notamment, dans les modèles sonores, de modéliser l'interaction de collision, et donc l'excitation par percussion.

- **Liaison Conditonnelle : module LIC**

Le formalisme définit une liaison non-linéaire plus générale (dont le module BUT est d'ailleurs un cas particulier) : la Liaison Conditionnelle LIC.

Un LIC repose sur un *automate d'états finis*. A chaque état E_i correspond un algorithme (généralement l'un des algorithmes précédemment exposés, mais il peut être différent). Le passage d'un état E_i à un état E_j est décidé par une *condition de transition*, les conditions de transition d'un état étant exclusives l'une de l'autre :

Algorithme

Passage (Ei Ej)

ssi (conditions sur les positions des <MAT>, les vitesses, les forces….)

La liaison LIC permet par exemple de rendre compte de tout comportement d'hystérésis. Ses utilisations sont cependant plus larges – nous y reviendrons dans le corps du document.

Modules d'interaction

Le formalisme CORDIS-ANIMA définit deux types de modules d'interaction, qui permettent aux modèles d'échanger une information bidirectionnelle avec l'extérieur (transducteur, fichier…).

Le premier est un module de type <MAT>. La position du module est élaborée en fonction de l'environnement externes à la simulation (elle correspond par exemple à la position d'un transducteur). La force appliquée au module par la simulation est disponible pour cet environnement.

Le second est un module particulier qui n'a qu'un point de communication de type L. La position qu'il capte (position du point M auquel il est connecté dans le modèle) est disponible pour l'environnement. Le signal de force entrant dans le modèle est, à l'inverse, élaboré à partir de données externe à la simulation (force exercée sur un transducteur).

Les algorithmes implémentant ces modules d'interaction sont divers. Nous décrirons plus loin les algorithmes des modules d'interaction utilisés dans GENESIS.

Ces modules supposent que la communication entre les modèles et l'extérieur est bidirectionnelle. Pour autant, parfois, le signal entrant dans la simulation - en provenance de l'extérieur - peut être constant. C'est le cas lorsque le module représente un transducteur sonore, qui n'est qu'un effecteur mais pas un capteur[1]. A l'inverse, le signal sortant de la simulation peut ne pas être utilisé. C'est le cas, par exemple, lorsque les valeurs entrant dans la simulation sont prises dans un fichier. Les modules ne réalisent plus, alors, une interaction, mais une communication. On dit qu'ils sont dégénérés.

[1] On pourrait envisager des transducteurs sonores bidirectionnels, susceptibles de renvoyer à la simulation une information sur l'état acoustique du lieu de diffusion, et donc de permettant un couplage *acoustique* entre la simulation et son environnement. Cela dépasse, cependant, le cadre de notre travail.

Modules fonctionnels : Modules de variation dynamique

On peut observer que certains instruments (au sens large) présentent des caractéristiques *permanentes*, alors que d'autres sont *modifiables dynamiquement*[1].

Pour représenter cette dernière possibilité il faut pouvoir modifier dynamiquement la structure du modèle. CORDIS-ANIMA définit pour cela les modules de modification dynamique structurelle et paramétrique.

- **Modules de modification structurelle dynamique MMS**

Le module de modification structurelle permet, en cours de simulation, de modifier la structure d'un modèle, c'est-à-dire de modifier la topologie du réseau <MAT>/<LIA> et d'ajouter ou supprimer des modules.

Dans son principe, le module de modification structurel se présente comme un point M ou L, dont le signal d'entrée (force ou position) permet de contrôler l'insertion ou la suppression d'une partie d'un modèle par l'intermédiaire d'une sortie booléenne (donc non physique).

- **Modules de modification paramétrique dynamique MMP**

Cette catégorie de modules permet de contrôler pendant la simulation les paramètres d'un ou plusieurs module(s) physique(s).

Un tel module se présente comme un point M ou L dont le signal d'entrée (force ou position) permet de moduler la valeur d'un paramètre d'un module physique, suivant des lois choisies par l'utilisateur.

Modules fonctionnels : Modules de modification du point de vue relatif

Enfin, CORDIS-ANIMA définit une dernière catégorie de modules fonctionnels : les modules de modification de point de vue relatif MPVR.

Insérés entre deux objets simulés ou entre un objet simulé et un transducteur, ces modules permettent d'appliquer des transformations sur les signaux échangés.

Cette transformation peut consister en une modification d'amplitude ; le module MPVR assure ainsi la mise en correspondance de deux univers d'échelle différente. La transformation réalisée peut être aussi plus complexe, comme dans le cas déjà évoqué où l'utilisateur interagit avec une simulation tridimensionnelle via trois touches du TGR© sur lesquelles un adaptateur mécanique est monté.

[1] L'ajout d'un timbre à une caisse peut être envisagée comme une telle modification dynamique.

1.3.8 - Implémentation de CORDIS-ANIMA ; moteurs de simulation CORDIS-Off et TELLURIS

Les moteurs de simulation assurent la mise en œuvre de premier niveau des modules définis dans CORDIS-ANIMA. Deux moteurs sont à l'heure actuelle développés.

Le moteur temps-différé CORDIS-OFF

CORDIS-OFF est le moteur dédié aux simulations en temps différé. Pour un tel processus de simulation, les calculs sont menés sans aucun contrôle d'horloge : les pas de simulations sont calculés le plus vite possible (compte tenu de la puissance de la machine), et les échantillons en sortie se présentent à une fréquence non contrôlée, voire variable. En conséquence :

- Aucune interaction n'est possible avec la simulation puisque le temps du calcul et le temps du modèle en mouvement ne sont pas les mêmes.
- Les flux en sortie ne sont pas disponibles en temps réel. Un fichier son et/ou un fichier film sont créés, et accessibles une fois seulement la simulation terminée. Les phénomènes générés par la simulation ne sont observés que lorsqu'elle est terminée.

CORDIS-OFF se présente comme une bibliothèque en langage C. Chacun des fichiers de la bibliothèque correspond à un module CORDIS-ANIMA, et comprend des fonctions d'allocation, des fonctions de connexion et des fonctions d'exécution.

TELLURIS

TELLURIS [Florens&al.98], [Giraud.99] est un programme autonome pour machines multiprocesseurs. Il est dédié au temps réel *dur* (c'est à dire impliquant un contrôle d'horloge strict) et intègre le traitement des multiples périphériques visuels, sonores et gestuels.

Ses fonctionnalités s'articulent autour d'un noyau, en charge des contrôles d'horloge et des échanges avec les périphériques. Le noyau confie à l'un des processeurs les tâches système pour lesquels le contrôle d'horloge n'est pas nécessaire, et isole les autres sur lesquels seuls les calculs de la simulation sont effectués.

Le moteur TELLURIS nécessite une technologie informatique complexe du fait des exigences fortes auxquelles il doit répondre. Il est encore à ce jour en cours de développement. C'est, d'ailleurs, l'une des raisons pour lesquelles nous avons décidé de ne pas l'utiliser dans les premières versions de GENESIS, qui reposent sur le moteur temps-différé CORDIS-OFF. C'est un choix essentiel, tant pour nos travaux que pour les possibilités offertes par GENESIS aux utilisateurs. Nous le motiverons en détail dans la partie IV, indiquant notamment pourquoi il n'a guère eu d'impact sur la définition des fonctionnalités de GENESIS.

1.4 - Environnements pour la création

Pour une mise en œuvre élémentaire de CORDIS-ANIMA et des simulateurs, l'ensemble constitué d'un langage de programmation et d'un compilateur est une base minimale. Il s'agit alors d'écrire pour chaque modèle un programme incluant la librairie de simulation. Cette démarche est couramment utilisée à des fins de recherches dans le laboratoire, notamment du fait de sa flexibilité.

Dans le contexte de la création artistique, toutefois, cet environnement minimal ne saurait être suffisant. Pour compléter et intégrer les périphériques – notamment gestuels – et le système CORDIS-ANIMA, il devient indispensable d'ajouter un troisième élément à l'outil de création : l'environnement.

L'environnement de création doit permettre une interactivité maximale : il est le lieu de la *médiation* entre l'outil et l'utilisateur et le support d'expérimentation de processus de création nouveaux. Il doit, entre autres fonctions, constituer une "représentation matérielle" de ces processus.

L'objet de nos travaux tient, précisément, à la conception de GENESIS, environnement dédié à la création musicale. Les fonctions, fonctionnalités, ergonomie et problématique attenantes seront affinées dans la suite du mémoire. Ce paragraphe propose, plus simplement, une présentation succincte des principaux environnements conçus au cours des années précédentes. Il clôture ainsi l'exposé du contexte dans lequel nos travaux se sont déroulés.

Si le principe d'un ensemble de périphériques permettant une interaction multisensorielle et CORDIS-ANIMA constituent une base commune, les environnements qui les intègrent diffèrent nécessairement suivant les pratiques artistiques qu'ils adressent. La création musicale et la création graphique, par exemple, supposent des types de modèles et des démarches différents[1]. Les environnements dédiés à chacune de ces formes d'expression seront, ici, abordés successivement.

1.4.1 - Environnement pour la création graphique

Le système ANIMA

Le premier environnement qui a été conçu au sein du laboratoire est le système ANIMA. Il résulte des travaux de thèse d'Aimé Razafindrakoto dans les années 80, sous la direction d'Annie Luciani [Razafindrakoto.86].

ANIMA reposait sur deux machines complémentaires : un mini-ordinateur LSI 11/02 pour les opérations de modélisation et un processeur vectoriel AP120 avec écran à balayage pour les simulations. Les deux machines échangeaient des données décrivant les modèles ainsi qu'un certain nombre de données résultant des simulations. Le modeleur était basé sur une représentation graphique interactive, soit symbolique, soit à visée plus figurative (figure 10). Bien que limité en puissance, le simulateur permettait de manipuler les objets en temps-réel *via* une première version du transducteur gestuel rétroactif. La visualisation des simulations faisait appel à un habillage élémentaire des éléments de liaison et des points matériels à l'aide de cercles et de segments (figure 11).

[1] Par exemple, là où pour le son on peut utiliser un nombre restreint de modules simulés à haute fréquence, on mettra régulièrement en œuvre pour l'image un nombre plus important de modules simulés à une fréquence beaucoup plus basse.

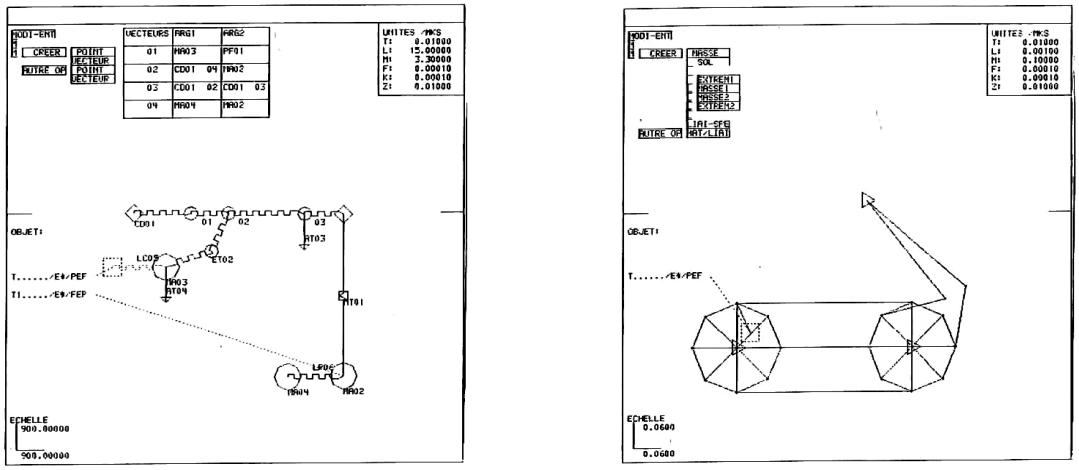


Figure 10 : représentation graphique symbolique (gauche) et figurative (droite) dans le modeleur du système ANIMA [Razafindrakoto.86].

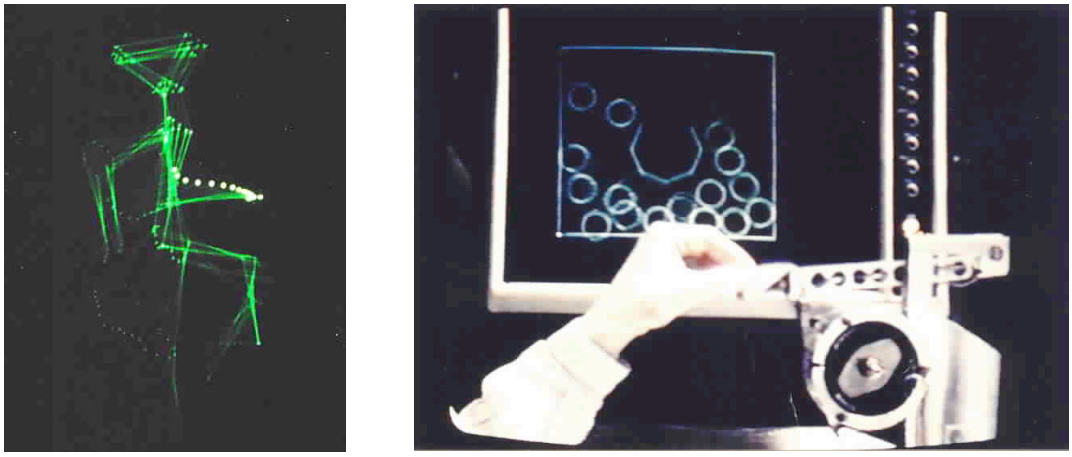


Figure 11 : objets simulés dans ANIMA

L'ébauche d'une nouvelle approche de la synthèse d'image animée a vu le jour avec ANIMA.

Dans l'approche traditionnelle, c'est la forme qui prédomine. Les modèles sont d'abord géométriques, puis mis en mouvement par des procédés plus ou moins automatisés[1]. Le processus de création s'organise alors avant tout autour de *l'image* (formes, texture, éclairage, etc.).

Dans le cas d'ANIMA, les modèles physiques particulaires sont au cœur du système. C'est au contraire la « physicalité » qui prime. Les images de synthèse sont produites par « *synthèse du mouvement et extraction de la forme* », plutôt que par synthèse de la forme et application du mouvement. Ainsi apparaît la notion de synthèse du mouvement, par opposition à celle de synthèse d'image, et en filigramme l'idée d'un *art du mouvement* encore en devenir.

L'approche suppose cependant qu'on sache habiller ou *mettre en espace* un ensemble de points afin de pouvoir procéder à leur visualisation. Elle conduit ainsi à de nouveaux

[1] Tels les techniques d'interpolation entre images (*Key Frames*), fonctions d'évolution, etc. Lorsque des « modèles physiques » sont utilisés (éléments finis, cinématique inverse, modèles particulaires, etc.), ils sont essentiellement envisagés comme un moyen pratique permettant de calculer les mouvements et déformations de formes conçues préalablement.

problèmes autour des notions de forme et de mouvement, de leur relation complexe et de leur perception.

Si ANIMA a mis en lumière la question de la *dualité forme-mouvement*[1], celle-ci est encore en grande partie non-résolues. Elle appelle notamment à une recherche sur les méthodes et techniques de mise en espace et à une réflexion sur les environnements de création. Les environnements pour la création graphique qui permettent une telle évolution imposent en effet au créateur de repenser son approche et, pour ce faire, supposent la conception de fonctionnalités jusqu'alors inconnues en image de synthèse. Plusieurs environnements ont été successivement élaborés dans le laboratoire. Le dernier en date, MIMESIS, tente de donner corps à l'ensemble des solutions jusqu'ici envisagées.

Le projet MIMESIS

Le projet MIMESIS, initié en 1995 sous la direction d'Annie LUCIANI, vise à offrir aux artistes un environnement graphique *puissant* et *complet* pour la création d'images animées [2].

MIMESIS s'organise autour de deux fonctionnalités ou modes principaux : MIMESIS-Méca et MIMESIS-Forme.

La partie MIMESIS-Méca (figure 12) est consacrée à la conception et à la simulation des modèles CORDIS-ANIMA. La description des réseaux MAT-LIA fait appel à un langage dédié et un compilateur. La spécification des paramètres et des conditions initiales des modèles est effectuée à l'aide d'une interface graphique. La simulation des modèles génère des fichiers .flm qui consignent les positions des MATs au cours du temps – il s'agit de synthèse du mouvement.

La partie MIMESIS-Forme permet l'habillage des nuages de points enregistrés dans ces fichiers. MIMESIS-Forme intègre un nombre étendu de primitives graphiques et implémente des techniques de mise en espace complexes, résultat de recherches autour de la problématique forme-mouvement ([Habibi.97] [3] [Guilbaud.02] [4] par exemple – figure 13).

---

[1] Comme la qualifie Annie Luciani.

[2] On remarquera incidemment le parallèle existant entre les objectifs de MIMESIS et ceux que nous adoptons pour nos propres travaux sur GENESIS. Les deux projets ont été initiés au sein du laboratoire à des moments similaires. Les recherches ont été menées en parallèle, ainsi que le développement des logiciels. Pour autant, ils posent des problématiques différentes.

[3] Harash Habibi montre l'intérêt d'avoir recours à des modèles physiques spécifiquement conçus pour l'habillage. L'écran d'épingle et le tapis d'aiguille sont deux exemples de tels modèles. Ils sont constitués d'un nombre important de modules en interaction, repartis sur une surface 2D. Lorsqu'on simule un autre modèle 2D, dit *modèle profond* et générateur du *mouvement*, il est possible de l'habiller en excitant l'écran d'épingle ou le tapis d'aiguille ; ceux-ci, alors, génèrent la *forme* qui sera visualisée à l'écran. Ces techniques ont été utilisées avec succès notamment pour la visualisation de modèles profond 2D de fumées.

[4] Claire Guilbaud a proposé plusieurs techniques pour la visualisation de modèles 3D très déformables.

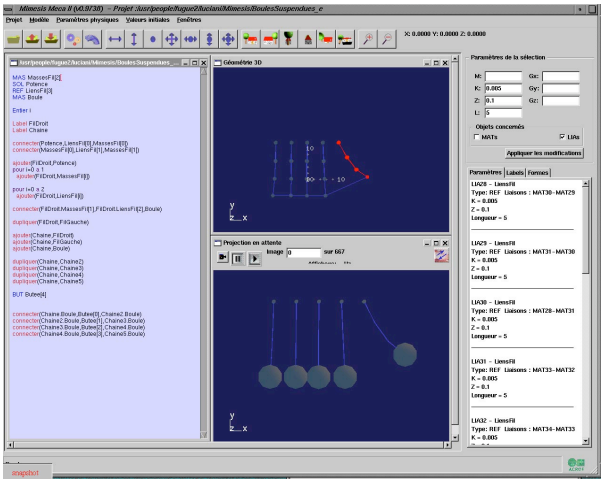


Figure 12 : capture d'écran de MIMESIS
on repère à gauche l'éditeur textuel pour la description des objets physiques, et au centre les fenêtres de prévisualisation.

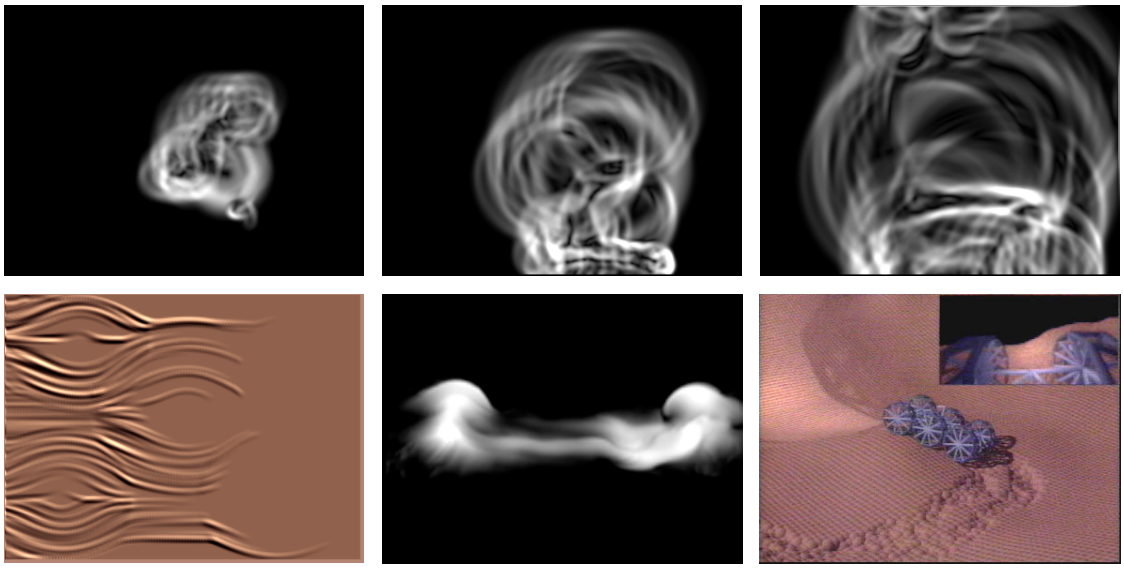


Figure 13 : images générées après habillage dans MIMESIS

1.4.2 - Environnement pour la création musicale

Trois prototypes d'environnement pour la création musicale ont été conçus et expérimentés au sein du laboratoire : la version 0 de GENESIS (Claude Cadoz / Olivier Raoult) en 1987, la version 1.1 (Claude Cadoz / Olivier Corbun, [Corbun.96]) en 1996, et enfin une ébauche d'une version 2.0 débutée en 1997.

Tout en éclairant chacun à sa manière la problématique d'un environnement pour la création musicale avec les modèles physiques particulaires, ces prototypes se sont avérés limités tant en ce qui concerne la qualité logicielle (stabilité, etc.), l'interactivité (ergonomie, etc.) que la puissance de modélisation (nombre de modules manipulables limité par le programme, choix ergonomiques malvenus, etc.). De plus, chacun ne

permettait d'aborder qu'une proportion très limitée du processus de création musicale complet avec l'outil - au sens ou nous l'envisageons plus loin dans ce document.

Ces prototypes auront constitué autant de propositions à analyser pour fonder notre réflexion. Toutefois, les décrire dès maintenant conduirait à être soit trop limitatif soit trop général. Ce n'est que dans la suite du document que nous détaillerons certaines des caractéristiques de chacun d'eux.

1.5 - Conclusion: Ici apparaît le paradigme de la simulation Multisensorielle Interactive d'Objet Physique

Nous avons donc exposé le formalisme CORDIS-ANIMA dans ses lignes majeures, en explicitant les raisons de sa genèse. Nous avons présenté les principaux modules tels qu'ils étaient définis lors du début de nos travaux, c'est-à-dire notamment dans [Cadoz&al.90a] et introduit les moteurs de simulation.

Ainsi exposé, CORDIS-ANIMA apparaît dès maintenant comme un *langage* et un *système de modélisation et de simulation d'objets physiques*[1].

Avec le TGR et en supposant suffisante la puissance des machines, les modèles respectant le système CORDIS-ANIMA peuvent tous acquérir un comportement physique perceptible par l'expérimentateur. C'est l'une des principales raisons pour lesquelles, en raccourcis, nous appellerons *objets* les modèles CORDIS-ANIMA.

C'est quoi qu'il en soit ici qu'apparaît le paradigme de la S*imulation Interactive et Multisensorielle d'Objets Physiques* (appelée plus simplement *simulation multisensorielle* dans la suite). Ce paradigme est au cœur de l'outil de création qui fait l'objet de nos travaux et plus généralement de la démarche du laboratoire.

Dans le chapitre suivant, nous tentons d'en appréhender les caractères innovants pour les processus de la création musicale.

[1] Les modèles physiques CORDIS-ANIMA sont souvent appelés modèles physiques *particulaires*, modèles physiques *à constantes localisées*, ou encore modèles *masses-interactions*. L'annexe A se propose de situer précisément le système de modélisation CORDIS-ANIMA en regard des autres techniques de modélisation physique aujourd'hui utilisées en Informatique Musicale.

Chapitre 2

NOUVEAUX OUTILS, NOUVELLES DEMARCHES

« La musique qui veut vivre et vibrer a besoin de nouveaux moyens d'expression et la science seule peut lui insuffler une sève adolescente »
Edgar Varèse, 1917 - [Hirbour.83].

GENESIS se doit d'ouvrir aux futurs utilisateurs tout le potentiel de créativité de la Simulation Intéractive et Multisensorielle d'Objets Physiques et les choix effectués dans sa conception vont être largement dominés par cet objectif. Cependant, il ne suffit pas de pressentir ce potentiel... Encore faut-il pouvoir le caractériser et en faire ressortir les points clé et, surtout, leurs conséquences prévisibles et nécessaires sur la démarche des utilisateurs.

Bien comprendre en quoi les démarches intellectuelles sous-tendues par la mise en œuvre de la *simulation multisensorielle* différent de celles qui ont été jusqu'à présent pratiquées a été pour nous d'une importance capitale. Il nous fallait franchir cette étape pour parvenir à ce que GENESIS mette en valeur ces différences et accompagne l'utilisateur dans l'évolution de son approche de la l'outil informatique pour la création.

Pour y parvenir nous avons voulu la situer par rapport à l'impact qu'ont eu les innovations technologiques du siècle sur les processus de création musicale. C'est cette démarche et surtout ses principaux aboutissements que nous relatons dans ce chapitre.

Le XX^{è} siècle a vu s'accomplir rien moins que deux « révolutions musicales »[1] de natures foncièrement différentes. La première est une révolution formelle ou esthétique. La seconde est de nature technologique. C'est elle qui nous intéresse particulièrement.

La musique n'a pas échappé aux révolutions techniques et scientifiques récentes. De nouveaux outils et de nouveaux moyens sont apparus et ont profondément transformé les conditions de la création.

Nous mènerons notre parcours parmi les faits marquants du XX^{è} siècle sous l'angle d'une part des mutations technologiques sous jacentes à ces outils, mais aussi d'autre part sous celui des nouveaux types de *dialogues* et de démarches intellectuelles qu'ils ont pu instaurer du fait de leur apparition. Il nous amènera à explorer successivement :

- Les instruments électroacoustiques ;
- L'enregistrement sonore ;
- La synthèse, l'analyse et la psychoacoustique ;
- La Composition Assistée par Ordinateur ;
- Les systèmes temps réel et les interfaces gestuelles ;

Nous serons alors armés pour discuter des apports essentiels de la *simulation multisensorielle* et du soutien que GENESIS devra leur apporter.

[1] Selon l'expression de Dominique et Jean-Yves Bosseur [Bosseur.93].

2.1 - Les instruments électroacoustiques

Risset distingue trois âges dans l'histoire de la "lutherie électrique" [Risset.99a]: l'âge de l'électrotechnique, celui de l'électronique[1], et celui du numérique.

L'age électrotechnique, au début du XXè siècle, a vu la naissance de nombreux *instruments électroacoustiques*[2]. Nous nous intéresserons plus particulièrement au cas du Theremin Vox[3].

Le Theremin est présenté au public en 1921 par son concepteur, Lev Sergeivitch Termen. A notre connaissance, c'est avec les Ondes Martenot l'un des seuls instruments du début du siècle dont le répertoire et la fabrication sont encore vivaces aujourd'hui. Dès les premières démonstrations, le son inouï du Theremin, pur et éthéré, fait sensation[4]. De fait, la première constatation importante relativement aux instruments électroacoustiques tient à la nouveauté des mondes sonores qu'ils apportent. Une seconde constatation relative à leurs interfaces gestuelles est peut être plus importante encore.

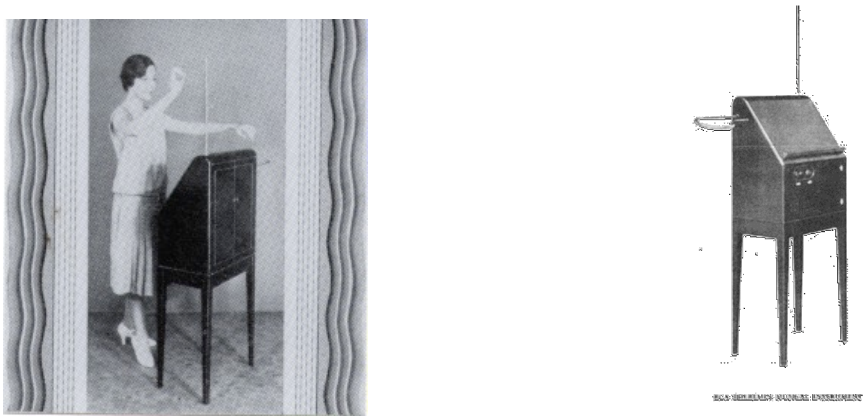


Figure 14 : le Theremin Vox.

Le succès du Theremin, en effet, réside probablement moins dans ses qualités sonores que dans son « geste instrumental » inédit.[5]. L'instrument est joué sans aucun

[1] Selon Risset, l'ère électronique débute avec la lampe triode, inventée en 1906 par Lee de Forest et permettant l'amplification des signaux électriques. Incidemment, on peut noter que Forest était motivé par un objectif sonore - d'ailleurs, il appelle d'abord son invention « audion » [Risset.99b].

[2] "On appellera électro-acoustique tout appareil qui par analogie avec les instruments acoustiques vibrant mécaniquement, produit des vibrations électriques, les utilise et peut les transformer en son". W.D.Kuhnlet - in [Barrière.90]. Le dynamophone de Thaddeus Cahill est sans doute, à l'aube du XXè siècle, le premier instrument électroacoustique. Impressionnante machine à son, il est constitué d'un ensemble de bobines capables de produire des signaux sinusoïdaux. Il sera, de plus, utilisé pour la première télédiffusion commerciale du son, via le réseau téléphonique (Telharmonium, 1895-1905). Plusieurs instruments électrotechniques puis électroniques lui succèderont (Sphaerophon (1926), onde Martenot (1928), Trautonium (1928) par exemple) qui utiliseront des procédés divers.

[3] Nous renvoyons pour de plus amples précisions à l'étude de Marc Battier, [Battier.99].

[4] Citons, en forme d'anecdote, les propos de Paul de Stoecklin, rapportés dans [Battier.99] : il déclare, enthousiaste, que le Theremin donne à volonté « l'illusion d'une corde vibrante (violon-violoncelle), d'une anche (larynx) ou d'un cuivre ». Aujourd'hui le son du Theremin nous semble au contraire bien loin de celui des instruments traditionnels. Les propos de Stoecklin prouvent combien il est ardu de caractériser un son nouveau et montre le poids de la culture de l'oreille musicale et illustrent la tendance naturelle à s'appuyer sur les timbres connus.

[5] Battier fait la même remarque en ce qui concerne les ondes Martenot. Il considère que le *contrôle gestuel* est la principale raison du succès ou de la faillite des instruments inventés à cette époque, qui pour certains d'entre eux reposaient sur les mêmes principes et, donc, disposaient de timbres équivalents [Battier.99]. L'ergonomie et la finesse de l'accès gestuel sont selon lui l'un des principaux garants de l'expressivité.

contact : la position des mains du musicien aux alentours d'une antenne modifie la capacité d'un circuit résonnant et contrôle ainsi la fréquence et l'intensité du son produit. L'annonce du concert qui a lieu en 1930 à New-York montre à quel point cet accès gestuel inédit a pu étonner :

> « Musique dématérialisée produite par des mouvements délicats de la main et des doigts, sans aucun contact réel avec l'instrument »[1].

Si le Theremin est une illustration frappante de la « *dématérialisation* », celle-ci apparaît commune à l'ensemble des instruments électroacoustiques. C'est que l'électricité autorise un découplage entre l'énergie mécanique et l'information. Avec elle, le travail (au sens de la Physique) de l'instrumentiste n'est plus corrélé à la puissance sonore, la fonction ergotique du geste musical disparaît. Il suffit d'un geste minimal, un geste de *contrôle*, pour déclencher une série d'évènements sonores.

Ces propriétés traverseront le siècle : orgue Hammond 1929 ; guitare électrique 1931 et plus tard, synthétiseurs et claviers numériques. Pour beaucoup, elles constituent une libération. Risset note ainsi qu'avec la genèse et le traitement électrique du son, le musicien « *s'affranchit* des *contraintes* de la mécanique » [Risset.99a]. Nous dirons que l'électricité a profondément modifié les conditions de la relation instrumentale en permettant de contourner, voire de dépasser, le corps à corps de l'homme avec son environnement.

Le Theremin, cependant, laissait entrevoir d'autres évolutions.

En 1929, dans la brochure publicitaire qui accompagne aux Etats-Unis la commercialisation de l'instrument par la firme RCA, on peut lire : "Tout le monde jouera le Theremin RCA de manière splendide. (...) Vous pouvez désormais tous jouer de la musique, et jouer de façon admirable, grâce à un instrument de musique superbement expressif, magnifiquement simple et *absolument non mécanique*. (...) Aucun élément matériel n'est nécessaire pour la production du son ni de la mélodie, *issue du cœur ou de la tête*"[2].

Une telle publicité est intéressante à deux titres. Elle révèle bien, tout d'abord, que la « non-mécanicité » est perçue comme une libération susceptible de rendre immédiate la musique en évitant l'apprentissage long et difficile des instruments traditionnels. Elle montre ensuite, par les propriétés exagérées qu'elle prête (dans un but commercial) au Theremin, que le mythe d'une musique en prise directe avec la pensée, d'une *musique de l'esprit* est très présent à l'aube du XX^{è} siècle.

Ce dernier point fait écho aux souhaits qu'exprimait Varèse dès 1917 :

> « Ce dont je rêve, ce sont de nouveaux moyens techniques qui puissent se prêter à n'importe quelle expression de la pensée et la soutenir (...). Je rêve les instruments obéissant à la pensée - et qui avec l'apport d'une floraison de timbres insoupçonnés se prêtent aux combinaisons qu'il me plaira de leur imposer et se plient à l'exigence de mon rythme intérieur. » Varèse, in [Hirbour.83].

Le souhait de Varèse restera présent tout au long du siècle : réduire - ou tenter de réduire - le chemin qui mène de la pensée du créateur aux oreilles de l'auditeur est l'un des

[1] "*dematerialized* Music Produced Solely by Delicate Movements of the Hand and Fingers in the Air, without Any Actual Contact with the instrument" (in [Battier.99]).

[2] Traduction en français de Marc Battier, [Battier.99].

défis que scientifiques et musiciens ont sans cesse relevé jusqu'à nos jours[1]. C'est ainsi que dans les décennies suivantes, on pourra assister parfois à l'abandon du musicien au profit de machines à produire ou reproduire le son[2]. Ce n'est alors plus seulement l'instrumentiste qui « s'affranchit de contraintes de la mécanique », mais la musique qui s'affranchit de l'instrumentiste. D'abord transformée, la relation instrumentale est maintenant éliminée.

Pour résumer cette tendance générale nous dirons qu'au cours du XXè siècle *la pratique musicale tend à se dématérialiser*. Le mythe tenace d'une musique de l'esprit[3], l'abandon de la fonction ergotique du geste, l'éviction du musicien, la musique sur bande, les outils numériques de composition et de synthèse en sont autant de manifestations.

2.2 - L'enregistrement sonore

Au tournant du siècle, au côté du son électrique figure selon Risset une seconde innovation technologique qui modifie profondément le rapport à la musique : l'enregistrement[4].

Les conséquences les plus évidentes de l'enregistrement tiennent aux conditions de la diffusion de la musique et au rapport nouveau qu'il instaure entre l'auditeur et l'œuvre. L'enregistrement permet à la musique d'entrer dans la sphère privée. Il est l'élément fondateur de l'économie de la musique que nous connaissons aujourd'hui, dans laquelle il génère les flux monétaires les plus importants.

Mais l'enregistrement est cependant à l'origine d'autres mutations qui concernent plus directement le processus de création musicale.

Il faudra un certain temps avant que les musiciens envisagent de dévoyer la fonction première de l'enregistrement qu'est la simple reproduction du son. Dès les années 1920, des expériences mettent bout à bout des portions de sons enregistrés, mais ce n'est qu'en 1939 qu'est créée la première *œuvre sur support* c'est à dire n'existant *que* sous forme d'enregistrement (John Cage, *Imaginary landscape n°1*). Elle inaugure une pratique qui va considérablement se développer par la suite, à tel point que l'œuvre sur support est devenue la forme la plus courante de la création musicale[5].

---

[1] Bien qu'elles soient encore à ce jour anecdotiques et qu'elles aient été critiquées, il est intéressant d'évoquer ici les expériences particulièrement symboliques menées entre 71 et 75 dans le but de commander la musique par la mesure des ondes cérébrale. Le "Cortical Art" (Roger Lafosse, Pierre Henry) en est une application – in [Risset99b].

[2] Le Theremin, à nouveau, laissait voir cette évolution future. Battier rapporte qu'après une représentation, l'humoriste La Fouchardière écrivait : "je me suis dit 'le chef d'orchestre est une mode qui a duré longtemps, mais qui finira par disparaître. Seuls resteront les musiciens qui remonteront du sous-sol ; et le public sera tout étonné d'avoir cru si longtemps à un mythe. O faillibilité ordinaires des prévisions humaines ! Ce sont les musiciens qui, fatalement, vont disparaître, anéantis par l'électricité. Et, seul, le chef d'orchestre restera à son poste, continuant à agiter les bras dans le champ magique des ondes. Il fera de la musique tout seul, sans musiciens" – in [Battier.99].

[3] Martin Laliberté nous rejoint sur ce point : « On retrouve depuis l'aube de la musique électrique les traces constantes du mythe de la machine sonore invisible qui répond idéalement à la pensée musicale – soit du compositeur, soit de l'interprète. (…) On cherche ainsi une immédiateté et une efficacité de la réalisation de l'idée musicale » [Laliberté.99].

[4] Le premier enregistrement sonore est réalisé en 1875 par Edison sur rouleau de cire (phonographe). Jamais auparavant on n'avait pu capter et réécouter un évènement sonore ; le son peut dès lors être isolé de sa cause originelle. La qualité de la reproduction augmente alors rapidement, notamment grâce à l'électricité.

[5] Ceci est vrai notamment pour la musique populaire, pour laquelle le concert - le "live" - est en règle générale une interprétation de l'œuvre préalablement fixée sur disque. L'enregistrement modifie également le concert lui même, avec la diffusion des œuvres sur bande – l'opérateur assurant seul depuis la console la

Une mutation profonde apparaît alors, consistant à envisager le son enregistré non plus comme une finalité mais comme un *matériau* - à travailler et à *composer* - et son support comme un *outil.* De nombreux compositeurs ont adopté cette approche de par le monde, mais c'est Pierre Schaeffer qui en est le principal théoricien. A partir de 1948[1] il se propose d'étudier systématiquement les conséquences tant esthétiques que méthodologiques de l'utilisation musicale du son enregistré allié à son support, et donne ainsi naissance à la *musique concrète*.

L'événement sonore était jusque là éphémère ; il s'évanouissait sitôt entendu. L'enregistrement permet de le présenter autant de fois qu'on le souhaite à l'oreille, condition nécessaire à l'émergence d'une notion essentielle : l'objet sonore.

> « L'objet sonore, écrit Schaeffer, c'est ce que j'entends, ce que je distingue ». Il y a objet sonore « lorsque j'ai accompli à la fois matériellement et spirituellement, une réduction plus rigoureuse encore que la réduction acousmatique : je m'en tiens aux renseignements fournis par mon oreille ; mais ces renseignements ne concernent plus que l'objet sonore lui-même ; je n'essaie plus, par son intermédiaire, de me renseigner sur autre chose (interlocuteur ou sa pensée). C'est le son lui-même que je vise, lui que j'identifie ».
>
> L'objet sonore, c'est le son hors de son contexte, le son en lui-même, dont l'intérêt et le caractère se définissent relativement à la perception et à l'écoute.
>
> « Enregistré, l'objet sonore se donnera comme identique, à travers les perceptions différentes que j'en aurai à chaque écoute ; il se donnera comme le même, transcendant aux expériences individuelles » [Schaeffer.66].

Pour Schaeffer, l'objet sonore est préalable à toute musique et toute organisation. Il se propose de rejeter « tout à priori musical », et défend une appréhension empirique du son et de la musique. C'est d'ailleurs dans cette attitude que la musique concrète trouvera son nom :

> « Nous avons […] appelé notre musique concrète parce qu'elle est constituée à partir d'éléments préexistants empruntés à n'importe quel matériau sonore, qu'il soit bruit ou musique habituelle, puis composé expérimentalement par une construction directe, aboutissant à réaliser une volonté de composition sans le recours, devenu impossible, d'une notation musicale ordinaire » [Schaeffer.66][2].

Le processus de création que propose Schaeffer est *concret* dans la mesure où le créateur part d'un matériau concret et préalable à toute structuration, qu'il manipule, transforme, combine ensuite par tous les moyens dont il peut disposer, pour aboutir à la forme de l'œuvre[3]. Avec la *démarche concrète* [1], la chose perçue a priorité sur la chose conçue.

---

diffusion sur un système électroacoustique complexe – et le principe des œuvres mixtes (*Déserts* de Varèse, 1954), qui font intervenir le jeu traditionnel et le son de la bande.

[1] L'histoire rapporte l'anecdote du "sillon fermé", selon laquelle c'est en écoutant un disque rayé à la TSF - la rayure est alors un défaut que les programmateurs s'efforcent d'éviter – que Schaeffer prend conscience du potentiel pour la création du son enregistré allié à son support.

[2] Manoury appelle cependant de ses vœux une notation étendue, propre à offrir au compositeur une possible écriture des sons dans toute leur complexité - voir [Manoury.99].

[3] Ce processus s'oppose à la démarche sérielle, et plus généralement à la situation traditionnelle dans laquelle le compositeur crée à l'aide d'un système de notations et de règles une partition -abstraite- qui ne deviendra tangible que lors de l'interprétation.

Force est de constater que l'attitude concrète née des innovations technologiques et du génie de quelques musiciens a profondément marqué les musiciens à la fin XXè siècle. Cette empreinte est allée bien au delà de l'impact de la musique concrète proprement dite : « qualité du son, vocabulaire des matières influenceront même les musiciens d'écriture instrumentale» (Michel Chion, in [Bayle&al.61]). Comme l'explique M. Robert, « Même les compositeurs qui n'ont pas goûté aux machines de studio reconnaissent qu'une nouvelle manière d'écouter s'est instaurée et cela pour toutes les musiques ; à partir de là se profile une nouvelle manière de 'communiquer par la musique' »[2].

Nous résumerons en trois points les mutations introduites par les propositions de Schaeffer sur les processus de la création musicale.

- Composition des sons, remise en cause de la forme

Après la musique concrète, les compositeurs ne composent plus seulement des signes. Ils peuvent aussi composer des sons . – et ce, qu'ils utilisent la bande ou qu'ils écrivent pour instrument traditionnels.

- Elargissement du « musical »

Tout son peut désormais être « musical ». Ce n'est pas là à proprement parler une nouveauté (voir par exemple [Russolo.13]). Mais l'enregistrement va permettre d'étendre définitivement l'univers des sons musicaux. Après Schaeffer, pour reprendre une formule de Molino, « le musical c'est ce que moi, compositeur, je décide de faire du sonore »[3].

- Modification de l'écoute

La notion d'objet sonore et l'écoute acousmatique se sont largement diffusées auprès des musiciens et des auditeurs. Le son a acquis *en lui-même* une richesse et une signification musicale accrues. L'oreille s'intéresse désormais à la *microstructure* autant qu'à la *macrostructure* de l'œuvre. Elle est plus que jamais sensible aux diverses qualités du son – situation qui se développera encore avec l'analyse, la synthèse et la psychoacoustique.

---

[1] Michel Chion résume ce que serait une démarche concrète *idéale* en 10 points [Chion.86]. Nous citerons ici leur en-tête, en rappelant toutefois qu'il ne faut pas y voir un *dogme* (que Schaeffer a toujours voulu éviter) mais plus simplement une présentation condensée des propositions de Schaeffer.

« 1 - Le compositeur concret va-et-vient du faire à l'entendre ». Il travaille avec le son, pas avec le signe. « 2 - Il est l'auteur des sons sur lesquels il travaille ». Il est réceptif à toutes sortes de caractères du son. « 3 - Il distingue complètement son et source sonore ». L'histoire du son ne l'intéresse pas. « 4 - Son champ de travail : le son enregistré ». « 5 – Le compositeur concret ne se laisse pas prendre aux mots de 'matériau' ou de 'manipulation' : pour lui, chaque son né d'un autre est un nouveau son ». « 6 – La création du son, dans l'œuvre concrète, se fait tout au long de sa composition ». « 7 - Il n'y a pas de sons naturels pour la musique concrète ». « 8 – Ni de trucage ». La « vérité » d'un son n'existe pas, tout son peut être valide quelle que soit son histoire. « 9 - En musique concrète, aux sources réelles se substituent les sources imaginaires ». « 10 – La musique concrète distingue entre ses moyens fondateurs et ses moyens accessoires ».

Ce dixième point appelle un commentaire. Le micro, le magnétophone et la console sont les *moyens fondateurs* de la démarche concrète, mais celle-ci reste particulièrement sensible aux innovations technologiques. Il ne fait aucun doute qu'on ne peut pas faire la même musique avec un disque microsillon, une bande magnétique ou un son numérisé. Le musicien concret a ainsi été friand mais aussi moteur d'importantes innovations technologiques. Dans cet ordre d'idée, on peut dire que « le traitement numérique des sons prolonge la musique concrète » [Risset.99a] : il laisse inchangés les principes fondateurs de la démarche concrète, mais décuple les moyens de représentation et de transformation du signal sonore.

[2] Robert, Martial : *Pierre Schaeffer : des Transmissions à Orphée. Communication et Musique en France entre 1936 et 1986* – Paris, L'Harmattan, 1999. Cité dans [Heinrich.2001], p 94.

[3] Molino, Jean : *la théorie d'hier à demain* , inharmoniques n° 89, IRCAM, Paris, 1991 - in [Heinrich.2001], p71. Notons que cet élargissement du 'musical' se constate autant dans la musique populaire que dans la musique savante.

2.3 - La synthèse, l'analyse, la psychoacoustique

La psychoacoustique[1] prend un véritable essor à partir des années 1960 avec l'emprise que permet l'ordinateur sur le signal sonore. En effet, l'approche scientifique du fonctionnement de l'audition et l'exploration des dimensions de la perception supposent d'une part une bonne connaissance de la microstructure des sons – accessible par l'analyse – et d'autre part une maîtrise parfaite de leurs paramètres – devenue possible par la synthèse. Réciproquement, en permettant de mieux cerner ce qui fait la richesse et l'identité d'un son, la psychoacoustique va pouvoir aider à conférer du caractère aux sons de synthèse [2].

Analyse, synthèse et psychoacoustique ont considérablement accru notre compréhension de la microstructure du son, tant physique que perçu, en permettant de le décomposer, de l'envisager comme un ensemble d'éléments analytiques ou générateurs. Leur influence sur le discours musical est particulièrement évidente dans le cas de la musique spectrale[3], mais leur impact est plus général.

Chez les musiciens, un nouveau vocabulaire s'est développé, ainsi qu'une nouvelle façon de penser le son musical, autour, par exemple, des notions de fréquences, de spectre, d'évolution spectrale, etc.

De nombreuses techniques de synthèse ont vu le jour depuis les années 60[4]. Le paradigme issu de la suite des programmes MUSIC initiés par Max Mathews[5] constitue à

---

[1] Rappelons que la psychoacoustique s'intéresse aux mécanismes physiques et cognitifs de la perception sonore. Elle vise notamment à mieux connaître les relations complexes entre paramètres physiques du son (fréquence, amplitude, spectre…) et paramètres perceptifs (hauteur, sonie, timbre…), ce qui fait dire à Heinrich qu'elle constitue un bipôle entre ces deux espaces [Heinrich.2001]. Elle étudie, en outre, l'organisation de la perception (capacités discriminatoires, reconnaissance de timbres ou de pattern musicaux, sensation d'espace…). Récemment, elle se penche sur les mécanismes complexes de la perception et de la cognition des *sons organisés*, notamment de la musique.

[2] La démarche méthodologique dite par d' *analyse par synthèse* proposée et mise en œuvre par Risset montre combien analyse, synthèse et psychoacoustique sont intimement liées. Avec elle, Risset a par exemple mis en évidence l'importance des évolutions temporelles du spectre pour l'identification d'un timbre donné, et permis les premières synthèses pertinentes des sons de cuivres.

[3] Le développement de la musique spectrale à Paris à partir des années 70 a été entre autre influencée par les travaux de JC Risset et J Chowning, comme en témoigne Dufourt. La musique spectrale donne une importance prépondérante aux *timbres* instrumentaux et à leur structuration dans l'espace temps-fréquence. Il s'agit de composer avec les timbres pour qu'en émerge l'harmonie, et d'intégrer en un tout forme et matériau. Ainsi, le compositeur Tristan Murail explique qu'au lieu « d'empiler des briques successives », il cherche à sculpter la matière sonore et à permettre de « voyager à l'intérieur du son et d'observer sa structure interne » - in [Heinrich.2001], p91. Ainsi « la musique dite 'spectrale', peut être considérée comme une projection à une échelle macroscopique de processus psychoacoustiques que l'informatique avait permis d'identifier à une échelle microscopique » [Dufourt&al.98]. Pourtant, chose remarquable en regard de ce lien de dépendance, la musique spectrale reste essentiellement une musique instrumentale. On voit là particulièrement bien comment les retombées des sciences et des technologies peuvent modifier intimement une pratique ancienne.

[4] Tels la synthèse additive, la synthèse soustractive, la synthèse FM, etc.

[5] Tels MUSIC I 1957, MUSIC III 1959, MUSIC V 1967, C-MUSIC, C-SOUND… A partir de MUSIC III, les programmes MUSIC sont modulaires ; leurs principes, incidemment, ont été repris dans d'autres domaines ultérieurement, par exemple dans des programmes de simulation électronique ou de traitement du signal comme BLODI, CSMP, etc. Les programmes MUSIC définissent un ensemble de modules élémentaires (« *building blocks* ») capable d'élaborer un *signal* en sortie à partir d'une entrée :

la fois un langage et un système. Il permet la mise en œuvre de la plupart de ces techniques, c'est-à-dire à la fois la définition des modèles, la synthèse et le contrôle de synthèses.

Avec le soutien des environnements de type MUSIC et de leurs successeurs, la synthèse numérique ne fait pas qu'étendre les possibilités de la génération électronique des sons (la « musique électroacoustique » des années 50). Elle ouvre bien de nouveaux univers sonores (qui ont d'ailleurs leurs limites, quoi qu'on ait pu en dire), mais c'est certainement le processus de création qu'elle autorise qui constitue son apport essentiel. La synthèse numérique offre en effet la possibilité d'une maîtrise complète du son par le compositeur.

Ainsi, on peut dire que la synthèse numérique porte la possibilité de *composer le son* lui-même, là où la démarche concrète permettait de composer des sons – ou des objets sonores – préexistants. Avec la synthèse, le compositeur peut « faire jouer le temps dans le son au lieu d'agencer des sons dans le temps », [Risset.99a]. La frontière entre substance et forme, entre son et organisation se réduit encore : le son est maîtrisé avec toute la précision requise pour une éventuelle écriture.

2.4 - La Composition Assistée par Ordinateur et les processus d'engendrement

Moyen de représentation par excellence, l'ordinateur s'est vu dès les années 50 confier la tâche de représenter la musique et de manipuler ces représentations. Hiller et Isaacson ont, les premiers, conduit en 1957 des expériences systématiques sur la composition avec l'ordinateur[1].. De l'avis général, leurs résultats sont plus intéressants sur le plan du processus mis en œuvre que sur celui du rendu musical. Ils préfigurent le courant de la composition automatique, emmené en France dès les années 60 par Barbaud[2] et dans une certaine mesure par Xénakis[3], aujourd'hui en grande partie abandonné, et plus récemment celui de la composition assistée par ordinateur (CAO), branche importante de l'informatique musicale.

---

oscillateurs, bancs de filtres, générateurs d'enveloppe, etc. Un instrument est une cascade de ces modules élémentaires dans un diagramme ou *patch*. Le patch comprend à son tour une ou plusieurs sorties ou voies et une ou plusieurs entrées ; il peut être réutilisé dans un *patch* plus important. A la notion d'instrument ou de *patch* s'ajoute celle de partition, ou *score*. Une partition est dans l'environnement MUSIC un ensemble de *notes*, spécifiées en termes de paramètres acoustiques (secondes, décibels et Hertz) ou plus traditionnels (note de la gamme, nuance…). La modularité est étendue à ce niveau, une partition pouvant être réutilisée comme une phrase dans une partition plus complexe.

[1] Leur première production, *Illiac suite for String Quartet* est une pièce instrumentale dont la partition résulte d'un double processus de génération puis de sélection. Des paramètres musicaux (hauteur, durée, intensité…) représentés par des nombres sont dans un premier temps générés avec une version de l'algorithme de Monté Carlo. Dans un second temps, ces « tirages » en grand nombre sont triés à l'aide d'un ensemble de règles musicales qui permettent d'évaluer la validité d'une séquence de données. Dans le cas du premier mouvement de la suite Illiac il s'agit des règles du contrepoint de première espèce implémentées à l'aide de chaînes de Markov. Les autres mouvements illustrent la composition stochastique et évoquent un style plus « contemporain ».

[2] L'œuvre *Factorielle 7* est présentée en 1960. Les travaux de Barbaud font largement appel à la théorie des ensembles [Barbaud.66]. Barbaud s'interdisait de corriger les produits de ses programmes. Lorsqu'il n'étaient pas satisfaisants, il préférait retravailler le programme lui même. Cette démarche est qualifiée *d'automatisme intégral*.

[3] Xénakis a notamment introduit les concepts de masses musicales, de musique stochastique et de musique symbolique et a formalisé le recours aux calculs probabilistes pour la composition de la forme musicale *et* la synthèse [Xénakis.63].

Définir précisément la CAO supposerait d'expliciter d'abord ce qu'est la *composition musicale,* tâche pour le moins délicate. Nous dirons, en résumé, que les recherches en CAO étudient l'ensemble des procédés et techniques permettant à l'ordinateur d'assister le compositeur dans son travail *d'organisation symbolique* du matériau musical, quelle que soit d'ailleurs la notation choisie pour représenter la musique[1].

L'étude de la CAO amène une constatation générale : « bien au-delà d'un gain quantitatif, [la CAO] opère une remise en question qualitative des modalités de la création » [Assayag&Cholleton.95]. Nous relèverons quelques-unes des mutations profondes (qui figuraient déjà en puissance dans les travaux de Hiller) induites par la CAO sur les modalités de création et sur le travail du compositeur.

Tout d'abord, notons l'importance croissante, dans la composition, des *procédés de génération automatique de forme.* De tels procédés existent certes depuis longtemps dans la création musicale[2] ; l'ordinateur permet cependant de les systématiser et surtout d'explorer les conséquences musicales du formalisme expérimenté.

Carlos Agon écrit à ce sujet : « Lorsqu'un ordinateur peut calculer très rapidement toutes les variantes combinatoires issues d'un formalisme donné, par exemple le formalisme sériel, la valeur musicologique ne vient plus simplement de ce principe formel, mais de son adéquation à la perception. […] La CAO a favorisé la démarche expérimentale qui permet de soumettre un très grand nombre d'instances musicales issues d'un formalisme au test de la musicalité. Plus encore, elle autorise l'expérimentation sur les formalismes eux-mêmes qui peuvent être testés et en retour modifiés ou abandonnés si ils ne remplissent pas leurs promesses » [Agon.98].

La CAO a donc ceci de fondamental qu'elle explicite ou *objective* le formalisme. Celui-ci n'est plus seulement dans l'esprit du compositeur. Représenté à l'aide d'un dispositif matériel (l'ordinateur) il peut être placé en vis-à-vis du compositeur. Comme l'écrit Cadoz, avec la CAO « un nouveau concept s'est formé, que l'on pourrait appeler le 'modèle matérialisé'. […] Le 'modèle matérialisé' est caractérisé par ses déterminismes propres, que le créateur définit, mais surtout par son autonomie, et il s'établit entre le créateur et le modèle une relation dynamique, un "dialogue" » [Cadoz.88a].

Avec la CAO, composer une œuvre ce n'est plus uniquement ou nécessairement écrire une partition – qu'elle soit composée de notes, de signes ou de mots. C'est être à même de définir – on peut dire véritablement *composer* – un système de règles, de le représenter dans la machine puis, le cas échéant, de choisir parmi les propositions que le programme peut faire. Comme l'écrit Cadoz, « la matérialité du dispositif, aux niveaux supérieurs de l'organisation, est une latitude nouvelle et féconde » [Cadoz.88a] puisque, entre le modèle matérialisé et le créateur, un dialogue d'un nouveau type est susceptible de s'établir.

---

[1] Cependant, la plupart des logiciels de CAO sont basés sur la notation traditionnelle, à laquelle ils ajoutent, à plus haut niveau, une représentation symbolique de l'organisation des segments musicaux.

[2] Il est certain que la musique occidentale a toujours entretenu une relation forte avec les Mathématiques. Le plus souvent, un compositeur utilise à la fois l'intuition (ou l'inspiration) et les calculs dans son travail - [Assayag, Cholleton.95].

2.5 - Les interfaces gestuelles, les systèmes temps réel

Depuis la conception du système hybride *Groove* par Max Mathews[1] dans les années 70, les systèmes musicaux numériques temps réels occupent une place croissante tant dans la communauté scientifique qu'auprès des musiciens. Les *nouvelles lutheries* ou *lutheries virtuelles* sont aujourd'hui l'un des axes de recherche les plus prometteurs en Informatique Musicale. Il suffit, pour s'en convaincre, de considérer l'importance des programmes qui leur sont consacrés.

Dans ce domaine, la norme industrielle MIDI a joué dès 1983 un rôle central et presque hégémonique, devenant le standard majeur de la communication de données musicales entre machines numériques. On peut certes mettre en question l'adéquation de la norme MIDI à la représentation numérique du geste musical[2]. Il est par contre incontestable qu'elle a largement contribué à l'essor des nouvelles lutheries et ainsi à la modification des principes anciens de la musique instrumentale. Aujourd'hui encore, l'essentiel des systèmes temps réels y a recours: « après échantillonnage, les signaux [issus du geste] sont le plus souvent formatés en informations de type contrôle MIDI » [Wanderley .99].

Selon Marcello Wanderley, un *instrument virtuel* peut, en règle très générale, être analysé sous trois angles :

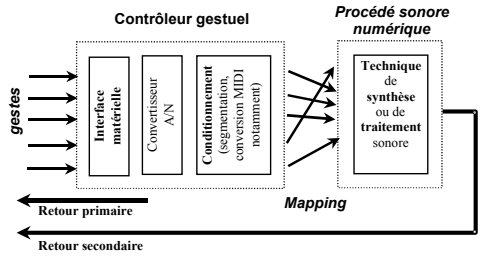


Figure 15 :
Constituants des « instruments virtuels »
d'après [Wanderley.99].

1 - les types de *gestes* qu'il permet ;

2 - le *contrôleur gestuel*, partie émergente de l'instrument responsable de son retour primaire[3] et chargé de convertir le geste capté en signal numérique, le plus souvent MIDI ;

3 - et enfin *lien instrumental* (ou « *mapping* »), partie logicielle responsable de la projection de ces signaux gestuels sur les paramètres de la technique de synthèse ou de traitement utilisée[4].

[1] Le système Groove est présenté en 1969 par Mathews et Moore. Il allie l'ordinateur à des synthétiseurs analogiques. La synthèse du son, encore impossible en temps réel avec le calcul numérique, est dévolue aux synthétiseurs. L'ordinateur permet une combinatoire et une édition de fonctions de contrôle f(t) pouvant être définies par une spécification numérique ou à partir de captures de gestes, puis assure leur conversion en tensions de contrôle ; il est ainsi responsable de la projection des gestes dans l'espace des sons. Le système Groove a entre autre donné naissance au principe du « conductor program », qui consiste à focaliser les interventions du musicien sur certaines dimensions sonores et musicales particulièrement signifiantes (tempo, équilibre de différentes voies…), le reste de la composition étant réalisé préalablement et enregistré dans la machine. Il a permis à ses concepteurs, plus généralement, d'imaginer une gamme étendue de contrôles gestuels, allant du geste élémentaire déclenchant un enregistrement au geste expert de l'instrumentiste.

[2] Beaucoup pensent que MIDI dénature le potentiel expressif du geste - voir [Risset.99a] par exemple. Ainsi, tout en offrant les possibilités qu'on connaît, la diffusion presque unilatérale de MIDI a probablement interdit ces dernières décennies une mise en œuvre optimale du geste musical de l'instrumentiste autour des technologies numériques.

[3] La rétroaction primaire, dans l'approche de Wanderley, consiste en l'ensemble des sensations qu'a l'instrumentiste au contact du contrôleur, du fait de sa matérialité : sensations visuelles, tactiles et/ou proprio-kinesthésiques, voire sonores si le contrôleur génère des sons (bruits des clés des contrôleur s'inspirant des instruments à vent, par exemple). Elle s'oppose au « retour secondaire » que constitue le son résultant des traitements numériques.

[4] Wanderley ajoute à ces dimensions la technique de synthèse ou de traitement utilisée. Nous l'omettons dans ce paragraphe consacré plus spécifiquement aux aspects gestuels.

Plusieurs tendances coexistent en ce qui concerne les contrôleurs et en conséquence, les catégories de gestes possibles. Nous en distinguerons trois :

- Une première approche consiste à munir un instrument traditionnel de capteurs additionnels. C'est le principe des instruments augmentés[1], qui permet au musicien de bénéficier de la maîtrise qu'il a acquise sur son instrument tout en augmentant ce dernier des capacités nouvelles du traitement ou de la synthèse.

- Une autre approche vise à ce que le musicien puisse effectuer des gestes inconnus de l'expérience instrumentale traditionnelle en le plaçant face à des dispositifs mécaniques entièrement nouveaux. En général, ces dispositifs sont conçus pour tirer parti de la « légèreté » qu'autorise l'électricité : « les différents modes de jeu que l'on peut alors obtenir se trouvent libérés de la résistance que la matière oppose au geste musical » [Toulemonde.99][2]. Parfois, ils permettent aussi un engagement plus complet du corps du musicien en proposant une interface corporelle et non seulement manuelle[3].

- La troisième approche se situe dans la continuité de cet objectif ; elle consiste à capter les déplacements et/ou mouvements du musicien sans qu'il n'ait plus à interagir avec une quelconque interface matérielle, à l'aide par exemple de capteurs répartis sur son corps ou d'une caméra dont on analyse les images[4]. Une situation nouvelle est ainsi créée, qui rompt avec la tradition du geste instrumental, jusqu'alors essentiel en musique, et laisse entrevoir une possible irruption de la danse dans l'univers des sons.

Le contrôleur gestuel détermine l'ergonomie du système et conditionne en grande partie la dextérité et la virtuosité. Mais il n'est pas seul responsable des possibilités expressives. Une fois les gestes captés, il faut encore « inventer le profil à phraser, […] donné à interpréter » [Pascal.99], c'est à dire déterminer le *lien instrumental.*

Avec la maîtrise du lien instrumental, timbre, situation dans l'espace ou encore dimensions formelles ou « compositionnelles » de l'œuvre deviennent, par exemple, accessibles au jeu[5]. Plus encore, le lien instrumental est susceptible d'être reconstruit à volonté, permettant au musicien de choisir la dimension du son qu'il va interpréter. Dès lors, créer ce peut être aujourd'hui *concevoir un accès gestuel nouveau au son*, proposer une *lutherie virtuelle* inédite et la placer au cœur même de l'œuvre.

---

[1] Il s'agit notamment de tous les instruments « midifiés ». Le nom d'Hyperinstrument, introduit par Tod Machover et Joe Chung, est parfois également utilisé pour les désigner.

[2] Citons pour exemple les tablettes graphiques, les molettes, la 'baguette radio', etc.

[3] Le Méta-instrument de Serge de Laubier est un exemple particulièrement frappant d'une telle approche [deLaubier.99] en capte simultanément trente-deux variables continues contrôlées par les bras, les mains mais aussi les jambes du musicien.

[4] L'installation « Mouve » de Laurent Pottier en est un exemple : les spectateurs déclenchent des évènements sonores par leur mouvement (marche, mouvement des bras, danse…) dans cinq zones de l'espace d'exposition.

[5] Risset note que « (...) la combinatoire et la présence de mémoires permettent de relayer l'instrumentiste dans ses fonctions de déclenchement et de commande et donc d'instaurer des modes complexes de contrôle en temps réel du rendu musical » [Risset.99a].

On espère des systèmes temps réels qu'ils permettent de réhabiliter la situation de concert que l'approche concrète et l'utilisation en temps différé de la synthèse avaient passablement transformée[1]. De façon plus générale, que ce soit en concert ou sur bande, les dispositifs temps réels sont censés conférer une vitalité aux sons que l'approche temps différé semble difficilement obtenir. Ils s'inscrivent ainsi dans un mouvement qui tend à une meilleure sensibilité des outils du musicien et à un meilleur contenu expressif des sons électriques, électroniques ou numériques. Si la synthèse a permis de « composer le son » numérique, on attend du temps réel qu'il le fasse *vivre*.

Reste que cette espérance est difficile à réaliser de façon satisfaisante, comme le reconnaît Wanderley :

> « En désolidarisant cause et effet, l'incursion de l'électricité dans la lutherie a profondément modifié la nature du jeu des instruments de musique. Et bien qu'une prodigieuse diversification des moyens de production sonore en ait résulté, ce changement de nature n'a pas encore engendré de stratégies de contrôle permettant de développer la même subtilité et la même ampleur de jeu que celle qu'on trouve dans les instruments traditionnels » [Wanderley.99].

Il devient alors essentiel de déterminer si la difficulté à obtenir tout à la fois virtuosité, sensibilité et musicalité est une caractéristique irrémédiable des systèmes temps réel ou si on peut espérer atteindre ces qualités par de nouveaux développements. L'approche de l'ACROE, en la matière, nous semble apporter quelques éléments de réponse.

[1] La *musique pour bande* étant devenue une forme obligée pour le compositeur désireux de se confronter aux possibilités sonores inouïes des nouvelles technologies. Michel Pascal pointe non sans raison l'importance de l'interprète dans la situation traditionnelle : « quelle écoute recherche le mélomane, l'amateur de musique classique : est-ce la partition du compositeur ou la prestation de l'interprète ? ». Selon lui, la fonction première du *live electronic* est de redonner « la plus grande importance à la performance de l'instrumentiste, de susciter cette qualité extrêmement particulière d'émotion, d'échange qui n'existe qu'avec le musicien face à son public » [Pascal.99]. Pour Serge de Laubier, l'enjeu est, de même, d'associer la « beauté éphémère du jeu instrumental et du concert » aux technologies du son numérique et de l'enregistrement, qui en était jusque là coupées.

Rappelons incidemment que d'autres courants musicaux qui ont recours à l'instrumentarium traditionnel donnent au XX^{è} siècle une place prépondérante à l'interprète (musique ouverte ou théâtre musical) et étendent ses possibilités (extension des modes de jeu sur les instruments traditionnels). Nous parlons ici, bien sûr, des créations qui ont recours aux nouveaux outils électriques et numériques.

2.6 - La simulation interactive et multisensorielle d'objets physiques

Face à ces récentes mutations des outils de la musique, nous situons les apports de la *simulation multisensorielle* sur quatre plans :

- Elle constitue une étape dans la recherche d'une meilleure sensibilité des outils numériques de la musique ;
- Elle rend possible un travail de *composition* du geste enregistré sous la forme de *trace gestuelle* ;
- Avec CORDIS-ANIMA, elle constitue une approche originale du *modèle physique*.
- Elle focalise l'attention sur le corps sonore et appelle le développement par le créateur d'une « *pensée physique* » ;
- Enfin, elle permet d'unifier dans un même formalisme les principes de génération du son et de la forme à travers la notion de composition d'objets physiques.

2.6.1 - Une étape du mouvement vers une meilleure sensibilité des outils

> « Il serait vain de croire […] que ce que le vivant a mis des centaines de millions d'années à élaborer, dans une interaction et une adaptation profondes avec l'environnement, peut être remodelé en quelques décennies d'informatique et quelques années de 'contrôleurs MIDI'. Le fonctionnement de nos muscles, de notre système de motricité ainsi que de notre système sensito-cognitif présente une base dont il convient de mesurer la stabilité. Pour qu'il puisse investir le nouvel espace que l'ordinateur lui ouvre, l'homme doit pouvoir y investir l'espace de son propre corps. Pour pratiquer 'les nouveaux gestes musicaux' il doit pouvoir retrouver ses gestes tout court » [Cadoz.99a].

Dans la situation traditionnelle l'expressivité instrumentale nécessite la *collaboration* de l'ensemble de nos aptitudes sensori-motrices. Elle résulte de l'interaction entre l'instrument et le corps de l'interprète, et émerge de la sublimation des contraintes qu'impose cette interaction.

Les paradigmes de la commande et du contrôle issus du traitement du signal apparaissent insuffisants pour reproduire cette situation. Certes, les nouvelles interfaces gestuelles et le *lien instrumental* qui réalise la projection du geste sur les paramètres de genèse et de transformation du son offrent des dispositions inédites. Mais, « s'affranchir des contraintes de la mécanique », c'est en vérité renoncer à une partie de notre potentiel expressif. Cette « mécanique » vécue parfois comme une contrainte est aussi une « contrainte créatrice ».

La simulation interactive et multisensorielle envisage la sensori-motricité de l'homme de manière globale. Avec elle, les modalités d'action et de perception de l'instrumentiste ne sont plus isolées mais mises en interaction cohérente par la simulation, chacune participant à une médiation étendue avec l'objet simulé.

Il ne s'agit pas cependant de recréer à tout prix une situation instrumentale traditionnelle mais, plus fondamentalement, de *pouvoir bénéficier du potentiel inédit de l'ordinateur sans renoncer à l'acuité de nos capacités naturelles d'action et de perception, et par-là même d'expressivité.*

La simulation interactive permet de *toucher la matière sonore* et d'interagir avec elle. En équilibre entre plusieurs nuances ou modes de jeu, sur le fil entre deux résonances, l'instrumentiste doit sans doute faire preuve d'une sensibilité importante pour résoudre correctement la *médiation* qu'impose l'instrument simulé. Il est *mis en danger*, bien plus qu'avec un geste de contrôle, mais comme le dit Michel Pascal cette mise en danger « est à la base d'une émotion spécifique donnée par l'interprète et contribue souvent fortement au résultat musical » [Pascal.99].

Elle permet aussi de conférer une meilleure *lisibilité* aux gestes. La lisibilité des gestes instrumentaux nous semble primordiale. Il ne suffit pas, en effet, de réintroduire l'instrumentiste « dans la boucle » ; encore faut-il que ses actions soient lisibles pour l'auditeur, c'est à dire qu'il soit possible d'inférer les gestes et l'interprétation en écoutant le son. La simulation d'objets physiques permet à la perception de distinguer efficacement les composants structurels et temporels des sons – c'est à dire de discerner dans le son à la fois l'instrument et les actions qui lui sont appliquées[1].

De façon plus générale, la simulation multisensorielle et interactive laisse espérer une relation instrumentale aussi intime que celle que nous connaissons avec les instruments traditionnels – moyennant, cependant, un important travail d'apprentissage.

A l'issue de ces quelques remarques, il apparaît que les périphériques à retour d'effort ne constituent pas seulement de nouveaux « contrôleurs » et CORDIS-ANIMA n'est pas seulement une technique de synthèse supplémentaire. Le paradigme de la simulation interactive et multisensorielle d'objets physiques constitue dans son ensemble une étape du mouvement qui vise à réintroduire l'homme, son corps et ses sens dans la pratique musicale et à conférer une meilleure sensibilité aux outils informatiques du musicien.

[1] Pour Michel Pascal la *lisibilité* du geste est rendue difficile avec les instruments numériques, du fait que « l'énergie physique ne sert plus à mettre en vibration des systèmes mécaniques ». Il ajoute : « S'il existait sur le marché des claviers au toucher physique rétroactif, ce serait un nouveau pas en avant, un nouveau type de geste, de toucher, à condition de pouvoir connecter cette force de rétroaction sur la réalité d'un contrôle du son et pas seulement sur une sensation physique du doigt » [Pascal.99].

2.6.2 - L'objectivation du geste musical ; la trace gestuelle ; composition du geste

Plusieurs auteurs considèrent que la composition musicale est indissociable de la notion de geste. Pour Risset par exemple, "plutôt qu'une description de la musique, les partitions sont un mode d'emploi, un codage des gestes à effectuer pour exciter l'instrument de façon à obtenir les effets musicaux recherchés" [Risset.99a]. Cadoz indique que « la composition sonore, pour partie, est une composition implicite et indirecte du geste instrumental » [Cadoz.88b].

La capture et l'enregistrement du geste musical sont courants en informatique musicale. La norme MIDI, par exemple, est souvent utilisée pour coder une séquence de gestes effectués sur un dispositif - mais d'autres formats plus spécifiques existent. Lorsque le geste est enregistré sous une forme quelconque il devient possible de le représenter, de le segmenter, de l'analyser, de le transformer – c'est à dire, en un mot, de l'*éditer* – avant de le réintroduire dans un dispositif de synthèse. Avec l'édition de geste s'ouvre un nouveau champ de recherche, mais aussi un nouveau champ de création. De la même manière qu'à l'enregistrement sonore correspond l'émergence de l'objet sonore, le geste, devenu objet, peut à son tour être susceptible d'une composition.

Le *geste enregistré* n'est cependant pas le geste lui-même. Quel que soit le dispositif utilisé, la capture du geste ne peut, en dernière analyse, qu'en offrir une *représentation* plus ou moins précise et opératoire. La question de la pertinence et de la portée des multiples approches aujourd'hui pratiquées se pose donc.

Dans la situation de CORDIS-ANIMA, le signal gestuel consigné sous forme de signal numérique résulte de la mesure du déplacement des pièces manipulées du transducteur gestuel. Ce n'est toujours pas le geste mais l'une de ses manifestations. Pour expliciter cette différence, Christophe Ramstein et Claude Cadoz ont proposé la notion de *trace gestuelle* [Ramstein.91], [Cadoz&Ramstein.90] : la trace gestuelle, avec la simulation multisensorielle d'objets physiques, c'est le signal capté au niveau du TGR, puis échantillonné et enregistré. Comparée à d'autres approches, la simulation interactive d'objets physiques se distingue alors à trois niveaux.

- Du fait de la précision du dispositif de mesure utilisé et de la quantité d'information contenue dans la trace gestuelle, celle-ci constitue un enregistrement particulièrement précis du geste – *haute fidélité* en quelque sorte. En comparaison, lorsqu'elle est détournée pour représenter des gestes, la norme MIDI les code sous forme évènementielle et n'offre qu'une représentation à un haut niveau d'abstraction.
- Du fait du retour d'effort et de la simulation d'une relation ergotique, la trace gestuelle est porteuse non seulement du geste effectué mais encore de l'interaction qui existait pendant le jeu entre l'instrumentiste et l'instrument simulé.
- Enfin avec CORDIS-ANIMA la trace gestuelle conserve sa signification dans le monde des processus informatiques. Elle *s'inscrit dans l'espace* lors de sa capture, mais encore pendant son édition et lors de sa ré-application. Cette propriété est de première importance pour le processus d'édition de traces : la trace, c'est certes un signal, mais c'est aussi une représentation opératoire et compréhensible des gestes.

Au sein de l'outil de création musicale que nous concevons, l'édition de traces gestuelles - enregistrement, analyse, modification, voire synthèse - peut être proposée comme une fonctionnalité majeure. La composition de traces devient alors une latitude nouvelle offerte au créateur. Elle permet l'émergence de la notion de « partition de gestes » ou de traces, ceux-ci conservant leur richesse musicale tout au long du processus.

2.6.3 - CORDIS-ANIMA, un système pour la modélisation physique

Le formalisme CORDIS-ANIMA est très souvent considéré comme une technique de modélisation physique.

De façon générale, les techniques de modélisation physique utilisées pour la création musicale se distinguent de la synthèse traditionnelle en ce qu'elles supposent de représenter et simuler non pas les phénomènes, mais les causes sous-jacentes qui le produisent et l'expliquent.

On peut s'interroger sur la validité d'une telle démarche.

D'abord, simuler un objet physique complet pour n'en garder qu'un phénomène sonore c'est à première vue mettre en œuvre une voie détournée. En effet, « (…) ce qui compte n'est pas le processus de production mais la perception, l'effet sur les sens, la sanction phénoménologique » [Risset.90]. Ensuite, puisque avec la synthèse on espère des sons *nouveaux* qui ne pourraient exister dans l'univers instrumental traditionnel, il est étonnant de vouloir modéliser la physique des instruments et donc de « chercher à reproduire ce qui existe déjà dans le monde acoustique" [Risset.90].

Les modèles physiques suscitent pourtant des attentes fortes, tant auprès des scientifiques qu'auprès des musiciens. Nous avons effectué une analyse critique des différentes techniques de modélisation physique aujourd'hui utilisées en musique, de la façon dont elles s'articulent entre elle et par rapport aux techniques de synthèse plus traditionnelles. Elle est proposée dans l'annexe A[1]. Cette analyse met bien en évidence les principaux espoirs suscités par l'arrivée des « modèles physiques » au sein des outils de la musique.

Tout d'abord les modèles physiques offrent la possibilité d'une synthèse particulièrement fidèle des sons d'instruments réels. Cela, cependant, n'est guère fondamental. Comme le rappelle René Caussé :

> « Utiliser ces modèles de synthèse lors d'applications musicales pour imiter les instruments "réels" n'a certainement pas grand intérêt et en parodiant G. Hegel on peut dire que développer de tels instruments est probablement 'un travail superflu... un jeu présomptueux. L'imitation ne pouvant produire que des Chefs-d'œuvre de technique, jamais des œuvres d'art' (Hegel, 1821, l'Esthétique) » [Caussé.90].

Fort heureusement, l'intérêt de la modélisation physique ne s'arrête pas là.

De façon plus générale, qu'ils modélisent un instrument réel ou qu'ils soient créés de toute pièce, les modèles physiques garantissent aux sons de synthèse une meilleure *plausibilité*[2]. La *plausibilité physique* des sons est une caractéristique essentielle : l'oreille, en effet, aime pouvoir inférer à partir des sons qu'elle entend, une cause physique possible, de sorte que « les timbres synthétiques ont plus de relief, ont une identité beaucoup plus

[1] Les principales techniques qui sont alors évoquées sont : l'approche par discrétisation, l'approche modale, l'approche source-filtre en interaction, la technique des guides d'onde et les systèmes particulaires auxquels se rattache CORDIS-ANIMA.

[2] On pourrait donc dire qu'un "certain réalisme" peut être recherché pour les sons de synthèse. "Réalisme" renvoie cependant à ce réel dont justement on veut par la synthèse s'affranchir. Nous préférons donc introduire la notion de *plausibilité physique*. Un son peut ne correspondre à aucune source acoustique, il peut n'en évoquer aucune, tout en étant *physiquement plausible.* Il est important qu'un son renferme un ensemble de caractères intimes, de liens dynamiques subtils entre paramètres acoustiques qui le rende *physiquement plausible*.

marquée, plus robuste, lorsqu'on peut imaginer de quelle manière ils auraient pu être produits physiquement » [Risset.90].

Qui plus est, le caractère *plausible* des sons générés par les modèles physiques présente est *robuste* face aux variations. D'une part leurs paramètres permettent souvent d'agir sur des dimensions du son pertinentes pour l'audition. D'autre part, ils tendent à réagir correctement, entendons de façon plausible, face à des sollicitations dynamiques. En quelque sorte, ils déplacent l'origine des subtiles variations dynamiques responsables d'une partie de la « vie » du son depuis le flux de contrôle (nécessaire avec les approches traditionnelles de la synthèse) vers le modèle lui-même.

Enfin, à travers les notions de *force* et de *position*, les modèles physiques permettent une projection particulièrement naturelle des signaux d'entrée – captés le cas échéant par des périphériques gestuels – sur les algorithmes. Associée à leur *robustesse*, cette propriété les rend particulièrement aptes à la mise en œuvre d'une situation de jeu qui permette un phrasé, une expressivité satisfaisante[1].

Ainsi, de façon générale, les techniques de modélisation physique utilisées en musique permettent des sons de synthèse plus intéressants et des possibilités d'expressivité accrues.

CORDIS-ANIMA ajoute à cela plusieurs propriétés qui lui sont spécifiques[2].

- Tout d'abord, les modèles CORDIS-ANIMA simulent le comportement vibratoire d'objets matériels, à l'inverse de modèles simulant certains phénomènes physiques impliqués dans la génération du son (comme la propagation par exemple) L'approche n'est en conséquence spécialisée ni dans une échelle de phénomènes dynamiques, ni dans une modalité sensorielle. Elle est par nature adaptée à la simulation de l'ensemble des phénomènes physiques pertinent à l'échelle des sens humains (et ce aussi bien pour l'œil que pour l'ouie ou la perception tactilo-kinesthésique).
- Ensuite, et ceci est essentiel, face aux objets modulaires définis par le formalisme, une représentation mentale pertinente et particulièrement naturelle, donc assimilable par le plus grand nombre est possible. CORDIS-ANIMA, favorise ainsi le développement d'une attitude intuitive face au processus de modélisation physique, qui se base non plus sur des considérations mathématiques ou scientifiques mais plus simplement sur notre expérience directe du monde.
- Enfin, CORDIS-ANIMA est fortement modulaire. Les modules CORDIS-ANIMA sont, à ce jour, les plus « élémentaires » qu'on puisse trouver parmi les techniques de modélisation physiques pour la musique. CORDIS-ANIMA conduit ainsi à son terme les principes de la modularité. Il peut être considéré comme l'homologue de ce que le paradigme des programmes MUSIC représente pour les approches traditionnelles de la synthèse.

[1] Constatons à ce stade que l'on a réalisé une boucle sur l'approche originelle du geste instrumental par l'ACROE. Dans le chapitre I, nous avons montré comment, partant du désir de réhabiliter la relation instrumentale, le laboratoire a introduit les périphériques spécifiques à retour d'effort puis le système CORDIS-ANIMA de modélisation physique. À l'inverse, les chercheurs intéressées par la synthèse sonore par modèles physiques constatent aujourd'hui que ceux-ci sont *effectivement* intéressants pour un renouveau du potentiel expressif des sons synthétiques, et que, par la métaphore naturelle des flux dynamiques qu'ils permettent, ils se prêtent particulièrement bien à un contrôle gestuel.

[2] Voir à nouveau l'Annexe A pour une comparaison plus circonstanciée de CORDIS-ANIMA aux autres techniques de modélisation physiques.

Au vu de ces différents points, CORDIS-ANIMA se présente non pas seulement comme une technique de synthèse qui permet de mimer la nature pour générer un son, mais bien comme un formalisme complet pour la modélisation physique, dont un créateur peut se saisir dans sa globalité. *Il n'est alors plus seulement utilisateur de modèles physiques, mais prend en main le processus de modélisation lui-même.*

Les objets (au sens général) qu'il manipule sont alors d'un nouveau type. Faire de la musique, ce n'est plus seulement agencer des notes, jouer d'un instrument, composer des sons ou par exemple élaborer des *modèles* de synthèse (signaux ou physiques). C'est *composer un ou des objet(s)* (ici au sens d'objets physiques), ou du moins leur représentation.

Ainsi, avec CORDIS-ANIMA le modèle physique utilisé dans une perspective musicale (non plus seulement *sonore*) est plus qu'une technique de synthèse. Comme l'explique Cadoz :

> « Bien qu'essentiellement attaché à la production du phénomène sonore dans les systèmes musicaux informatiques, [le modèle physique] annonce véritablement une mutation dans la manière même d'envisager le rôle de l'ordinateur dans le processus général de représentation et de création » [Cadoz.90b].

2.6.4 - Une musique des corps sonores ?

« Le corps sonore, touché mécaniquement, s'éveille à la vie, manifeste son existence ou plutôt son organisation, et parvient à notre connaissance *»* [Schaeffer.66].

Avec CORDIS-ANIMA, l'objet physique ou *corps sonore* est donc au centre du processus de création. Bien qu'il soit virtuel, c'est lui qui est façonné, modifié, travaillé. Puis il « s'éveille à la vie » lors de la simulation, imposant sa réalité à notre perception et permettant le cas échéant l'expressivité de l'instrumentiste[1].

En plaçant l'objet sonore en vis-à-vis du compositeur, la démarche concrète a permis l'émergence d'une nouvelle musique. Nous pensons qu'à son tour, en focalisant l'attention sur le *corps sonore*, le paradigme de la simulation d'objets physiques propose une nouvelle façon de *faire* la musique. De là, peut-être, pourrait naître également une nouvelle musique, construite autour de ce qu'on pourrait appeler la *poésie des corps sonores*. Cette vision s'appuie sur quelques constats :

- Avec la simulation d'objets physiques, la conception même de l'objet, le jeu expert du musicien avec cet objet devenu instrument et la composition (ici au sens traditionnel) pour cet instrument sont du ressort et de la responsabilité du seul utilisateur. D'une certaine manière, ce dernier cumule les métiers autrefois partagés par le luthier, l'instrumentiste et le compositeur. Ce faisant, la simulation interactive et multisensorielle d'objets physiques réintroduit au cœur de l'environnement audio-

[1] Notons que dès lors l'utilisateur de la simulation interactive d'objets physiques devrait pouvoir à tout instant se faire une représentation aussi efficace que possible de ce corps sonore qu'il manipule ou sur lequel il joue. Nous avons dès la partie I évoqué les périphériques qui permettent de présenter comme "réel" l'objet simulé ; pour autant, cette "impression de réalité" du corps sonore virtuel devrait être accessible tout au long du processus de création, notamment pendant la conception. Soutenir cette "impression de réalité" du corps sonore est donc l'une des fonctions essentielles de l'environnement GENESIS. On voit ici émerger le problème de la *représentation* des objets au sein de GENESIS, sur lequel nous reviendrons avec précision.

numérique cette notion de corps sonore dont l'importance avait été atténuée au siècle précédent[1].

- Ce processus s'accompagne de possibilités nouvelles du fait du support informatique. L'ordinateur permet d'ausculter l'objet virtuel, de le mesurer, de l'analyser, d'en comprendre et d'en affiner le fonctionnement comme jamais. Ainsi ce n'est pas seulement un renouveau de la poésie du corps sonore que le musicien tient en puissance, mais encore la possibilité d'une exploration à l'intérieur même de ce dernier[2].

La simulation offre de plus une articulation nouvelle entre réel et imaginaire, autour de la notion de *plausibilité* des objets simulés. Elle permet d'évoquer un objet réel sans lui être pour autant équivalent et ainsi d'envisager la construction d'un discours musical autour de la similitude et de la différence. Elle permet également de simuler des objets inouïs, défiant l'imagination et les possibilités technologiques, mais qui restent ancrés dans le réel par la robustesse du formalisme et la cohérence des phénomènes multisensoriels générés.

2.6.5 - Continuité de la forme au matériau et unification des processus de génération : le modèle physique, métaphore pour la composition

Il existe encore aujourd'hui une frontière nette entre les procédés qui permettent la génération du *matériau sonore* et ceux qui visent la genèse de la *forme musicale*. Cette frontière se dresse aussi bien au niveau des outils - avec d'un côté des outils pour la synthèse de son et de l'autre des outils d'aide à la composition - qu'au niveau des processus de la création musicale - qui opposent généralement la phase de genèse du matériau sonore à celle qui consiste à l'organiser[3].

Les récents travaux de Claude Cadoz tendent à montrer que le paradigme de la simulation d'objets physiques permet de dissoudre cette frontière [Cadoz.02a], [Cadoz.02b][4]. Il autorise dans le même temps l'engendrement du matériau musical et celui de la forme musicale. Les objets physiques simulés, en effet, ne sont pas limités par principe à l'échelle des phénomènes sonores audibles. On peut par exemple avec CORDIS-ANIMA construire des objets dont les constantes de temps sont arbitraires. Avec un objet basse-fréquence et un corps sonore en *interaction* par exemple, on génèrera à la fois le son

[1] On se rappellera l'importance du corps sonore pour le musicien concret : il est constamment à l'écoute des objets qui l'environnent, et en recherche de nouveaux corps aux sons inouïs, ou de nouvelles techniques pour donner à entendre autrement leurs vibrations. Cependant, la démarche concrète suppose que la causalité physique soit rapidement masquée pour que puisse émerger la notion d'objet sonore. Dans l'approche concrète, le corps sonore est écarté du processus de création dès que le son est enregistré.

[2] Incidemment, ce point permet de faire de GENESIS un environnement pédagogique d'apoint pour l'apprentissage de l'instrumentation, ce qu'Arnaud Petit, responsable de la classe de composition du Conservatoire National de Région de Grenoble, a eu plusieurs fois l'occasion d'expérmienter. A. Petit indique que la confrontation à GENESIS développe chez les étudiants-compositeurs une compréhension intuitive (ou 'sensible') des phénomènes physiques dont les instruments de musique réels (notamment à corde ou à plaque) sont le siège. Elle affine leur 'écoute intérieure' et leur permet, plus généralement, de mieux appréhender l'univers instrumental. Selon A. Petit, la mise en œuvre de GENESIS dans un contexte pédagogique constitue un intérêt fondamental du logiciel et de CORDIS-ANIMA.

[3] Notons cependant qu'il existe des tentatives d'intégration de ces deux aspects dans des logiciels tels qu'OpenMusic [Polfreman.02]. Mais dans ces logiciels des principes différents sont mis en œuvre pour la genèse du son et de la forme, de sorte que la frontière demeure.

[4] Une grande partie de ce paragraphe puise ses sources dans les réflexions actuellement menées par Claude Cadoz et les résultats expérimentaux qu'il a obtenu ces dernières années. Une réflexion plus poussée les concernant pourra être trouvée dans ces deux références.

et des évènements sonores, identifiables comme tels. On peut imaginer que cet objet générateur soit, à son tour, soumis à un autre objet qui le contrôle, et ainsi de suite jusqu'à ce qu'un *objet complet*, sorte d'automate musical maîtrisé par le compositeur, soit capable d'engendrer une séquence musicale arbitrairement longue et complexe de par sa structure même - et ce jusqu'à une œuvre complète.

Cependant, on s'aperçoit que l'unification des principes de la génération du matériau et de la forme au sein d'un même paradigme masque des apports encore plus fondamentaux.

- La composition d'objets

La composition, au sens traditionnel, consiste à agencer des évènements musicaux pour construire une forme musicale. En ce sens, la notion *d'objet complet* montre que la simulation d'objets physiques permet de *composer*. Pour composer, cependant, il ne s'agit plus d'agencer des notes, des signes ou des sons, mais, ce qui est nouveau, de mettre en interaction des structures physiques.

Composer, avec CORDIS-ANIMA, c'est « com-poser » des objets, c'est à dire, au sens littéral, les *poser ensemble.* Ce processus de *com-position* amène à envisager les modèles non plus seulement comme des objets générateurs de son, mais permet également de considérer CORDIS-ANIMA comme un système de notation de la musique à la fois dans ses dimensions sonores et formelles. Les *modules physiques*, symboles musicaux radicalement nouveaux, s'ajoutent aux signes de la musique que le XX^{è} siècle a déjà considérablement élargi. En suivant la pensée de Manoury [Manoury.99], ils portent en germe, peut-être, des idées musicales inédites.

- L'interaction entre échelles

Une disposition nouvelle et féconde apparaît du fait de la possible collaboration entre des objets évoluant à différentes échelles de temps.

L'interaction entre objets implique une rétroaction des objets haute fréquence sur les objets basse fréquence qui les excitent. Elle est responsable, dès lors, de micro-variations dans l'organisation des évènements générés sur l'objet haute fréquence, lesquelles peuvent s'avérer musicalement pertinentes. L'interaction entre objets permet une vitalité sonore qui serait inaccessible si on générait les évènements dans un premier temps – en les organisant dans une quelconque partition – pour les traduire dans un second temps à l'aide d'un procédé de synthèse.

Le principe de la composition d'objets et la maîtrise d'une possible interpénétration des échelles de temps sont des dispositions nouvelles, dont plusieurs expériences et création récentes ont pu montrer la portée[1].

[1] Claude Cadoz y a largement recours dans la pièce Pico..TERA, présentée en création mondiale lors des Rencontres Informatique et Création Artistique qui se sont tenues à Grenoble en 2001. Cette pièce résulte de la simulation d'un unique objet CORDIS-ANIMA, qui comprend des structures sonores, basses fréquences et très basse fréquence en interaction. La spécification d'un état initial sur cet objet permet, même si il est simulé de façon complètement isolée, de générer à la fois le son et la forme de la pièce sur une durée de 5 minutes. Pico..TERA, de par sa qualité musicale, apporte à nos yeux la preuve de l'intérêt de CORDIS-ANIMA en tant que paradigme de génération englobant et formalisme transversal pour la composition [Cadoz.02a], [Cadoz.02b].

CONCLUSION : PENSER PHYSIQUE

On attend des modèles physiques des sons plus intéressants, plus *plausibles* ou plus riches. On leur demande des paramètres plus naturels et des flux de contrôle dont les effets sont plus « parlants » à la perception. Mais, pour notre propos, là n'est pas l'essentiel.

Les modèles CORDIS-ANIMA établissent un parallèle entre l'univers réel et celui des processus informatiques. La Simulation d'Objets Physiques, lorsqu'elle est Interactive et Multisensorielle, permet un continuum entre ces deux univers. L'ensemble tend à amener « l'ordinateur musical » à la mesure de l'homme.

La Simulation Interactive et Multisensorielle d'Objets Physiques se place ainsi en rupture. Aux notions de commande et de contrôle, elle oppose celles d'interaction et de corps à corps. Au signal de commande gestuelle elle substitue la trace, qui depuis sa capture jusqu'à son application, en passant par son édition, s'inscrit dans l'espace et conserve sa « richesse vivante »[1]. A l'idée que le substrat de toute musique est le son, elle répond que le corps sonore le précède. Elle marque ainsi le *retour du corps matériel* dans la musique : le musicien d'abord, mais aussi la notion d'instrument et plus généralement celle des objets physiques qui interagissent, qu'on manipule et qui résistent.

Envisagés sous cet angle, les modèles physiques sont susceptibles de devenir la base d'un nouveau langage musical dont les « objets » seraient les nouveaux symboles.

Toutefois, pour être pratiqué, ce langage nécessitera une mutation profonde dans les démarches de la création musicale. Il appelle à la mise en œuvre d'une réflexion tournée vers les causes plutôt que vers les effets. Une approche dans laquelle la notion d'interaction tient une place essentielle. Une *pensée physique,* en quelque sorte.

Il reste que les approches du modèle physique n'offrent pas toutes le même soutien pour opérer cette transformation.

Modulaire, basé sur des « briques élémentaires » simples, compréhensibles et perceptibles, pouvant s'accompagner d'une représentation mentale pertinente et opératoire des algorithmes en jeu, CORDIS-ANIMA est en la matière particulièrement prometteur.

Mais CORDIS-ANIMA reste un formalisme qui n'est pas praticable tel quel sans un environnement qui lui soit spécifique.

Le rôle de GENESIS se dessine alors clairement.

GENESIS devra apporter un ensemble de fonctionnalités pour la mise en œuvre de CORDIS-ANIMA. Il lui faudra mettre en valeur ses spécificités en l'environnant de nouveaux concepts propres à renforcer son potentiel pour la création. Enfin, placé en interface entre ce formalisme et le musicien, GENESIS aura pour mission primordiale d'accompagner celui-ci dans l'évolution de son approche de la création musicale. Son ambition est de devenir le lieu du dialogue où l'œuvre est inventée.

[1] [Cadoz.88b]

Partie II

Interaction Homme-Machine
et méthodologie de conception de GENESIS

> The real problem with the interface is that it is an interface. Interfaces get in the way. I don't want to focus my energy on an interface. I want to focus on the job.[1]
>
> Donald A. Norman
> *Why Interfaces Don't Work*
> [Norman.90

[1] Traduction de l'auteur : « le véritable problème que pose l'interface est qu'elle est une interface. Les interfaces s'imposent. Je ne veux pas consacrer mon énergie à l'interface. Je veux la consacrer à la tâche ».

Un bon outil doit avant toute chose être *utilisable* – ce que, à propos des interfaces, le psychologue cognitif Donald Norman traduit en ces termes :

> « Le premier principe pour la conception des interfaces, que ce soit pour une poignée de porte ou pour un ordinateur, est de garder à l'esprit qu'un être humain veut les utiliser »[1] Norman, in [Rheingold.90].

Ce principe est on ne peut plus simple. Sa mise en œuvre, par contre, ne l'est pas nécessairement. La conception d'un *outil informatique* appelé, comme GENESIS, à être « utilisé par des utilisateurs » ne saurait être abordée sans que les conditions mêmes de l'interaction homme-ordinateur ne soient sérieusement considérées.

L'Interaction Homme-Machine (IHM) et l'Ergonomie[2] sont des domaines multidisciplinaires. Ils s'appuient sur les compétences complémentaires de l'informaticien, de l'ingénieur, du psychologue, de l'ergonome, du graphiste, de l'économiste, etc.

Le présent travail ne prétend ni faire une synthèse exhaustive des questions posées dans ces domaines complexes ni y apporter une contribution. Plus humblement, son objet est d'exposer les éléments technologiques, scientifiques et méthodologiques qui ont structuré notre travail sur l'interface.

Le chapitre 3 présente quelques considérations générales sur les interfaces et les fondements scientifiques et technologiques de l'IHM.

Le chapitre 4 résume les principes et guides sur lesquels nous nous sommes au jour le jour appuyé dans notre travail.

Le chapitre 5, enfin, conduit de ces considérations théoriques à l'exposé de la méthodologie adoptée pour concevoir les fonctionnalités et l'ergonomie de GENESIS et ouvre, ainsi, à la suite du document.

---

[1] Traduction de l'auteur. « *The first principle of human interface design, whether for a door knob or a computer, is to keep in mind that the human being want to use it* ».

[2] Selon la Société d'Ergonomie de langue Française, « l'ergonomie est la mise en œuvre de connaissances scientifiques relatives à l'homme, et nécessaires pour concevoir des outils, des machines et des dispositifs qui puissent être *utilisés avec un maximum de confort, de sécurité et d'efficacité* pour le plus grand nombre » - adoptée en 1988.

Introduction

Contexte et bibliographie

Partie I : le paradigme de la simulation interactive et multisensorielle d'objets physiques

Partie II : Interaction Homme-Machine

Chapitre 3 Interface, Homme, Ordinateur 64

3.1 - INTERFACE, SYSTÈME INTERACTIF 64
3.2 - L'HOMME DANS SON ENVIRONNEMENT 67
3.3 - L'ORDINATEUR « COMMUNIQUANT » 78

Chapitre 4 Guides et Principes ergonomiques 89

4.1 - LES PRINCIPES INSPIRÉS DE HANSEN. 89
4.2 - MODÈLE MENTAL, *AFFORDANCE*, CONTRAINTE, *MAPPING*, VISIBILITÉ ET RETOUR : L'ANALYSE DE NORMAN 91
4.3 - MÉTAPHORES 94
4.4 - ECONOMIE, SIMPLICITÉ ET SOUCI DU DÉTAIL 96
4.5 - COHÉRENCE ET CONFORMITÉ 97
4.6 - QUELQUES CONSEILS POUR UNE BONNE COMMUNICATION VISUELLE 98
4.7 - OBJET D'INTÉRÊT ET MANIPULATION DIRECTE : « MOINS D'INTERFACE POUR PLUS D'INTERACTION » 100
4.8 - THÉORIE ET PRINCIPES *VS* PRAGMATISME ET COMPROMIS 102

Chapitre 5 Méthode pour la réalisation de GENESIS 104

5.1 - PREMIÈRE ÉTAPE : ANALYSE DES TÂCHES 104
5.2 - DEUXIÈME ÉTAPE : PROPOSITIONS, DÉCISION 107
5.3 - TROISIÈME ÉTAPE : IMPLÉMENTATION ; GÉNIE LOGICIEL 109
5.4 - EVALUATION ET TESTS 111
5.5 - DÉMARCHE INCRÉMENTALE ET ITÉRATIVE ; LES BOUCLES IMBRIQUÉES 117

Conclusion 120

Partie III : Le processus de création avec GENESIS. Analyse des Tâches

Conception de GENESIS

Partie IV : Choix généraux pour l'environnement GENESIS

Partie V : Fonctionnalités fondamentales de l'atelier de lutherie GENESIS

Partie VI : Fonctionnalités évoluées du mode lutherie

Partie VII : Simulation et Phénomènes

Conclusion et perspectives

Chapitre 3

INTERFACE, HOMME, ORDINATEUR

3.1 - Interface, système interactif

3.1.1 - Définition

Une interface est une surface qui sépare deux milieux et à travers laquelle ils échangent et interagissent. Comme l'explique Brenda Laurel [Laurel.90a], pour permettre l'interaction, l'interface doit synthétiser certaines des propriétés des deux milieux et médiatiser auprès de chacun d'eux certains aspects et actions de l'autre.

Un exemple typique d'interface est la *poignée de porte*. La poignée se présente comme un intermédiaire entre la porte et l'homme. Ses propriétés tiennent bien compte à la fois de la porte (sa position est opposée aux gonds, elle est fermement attachée, sa rotation déclenche la gâche, etc.) et de l'homme (sa forme est adaptée à la main, son angle de rotation correspond à celui de poignet, etc.). La poignée, enfin, médiatise auprès de l'homme les actions possibles sur la porte, et *invite* à ce qu'elles soient réalisées : déclencher la gâche, faire pivoter la porte par exemple. De la même manière, elle médiatise (ou adapte) vers la porte les actions entreprises par l'homme.

Dans le cas de l'interaction homme-ordinateur, le concept d'interface s'applique à deux niveaux. L'interface, c'est ce qui sépare matériellement notre corps de l'ordinateur, c'est à dire les périphériques. Mais c'est aussi la partie du programme concernée par l'interaction avec l'utilisateur (figure 1). C'est ce second aspect, généralement désigné par les termes d'*interface utilisateur*, qui nous intéresse ici plus spécifiquement : GENESIS *dans son ensemble* peut ainsi être envisagé comme une interface entre CORDIS-ANIMA et les moteurs de simulation d'une part, l'utilisateur d'autre part.

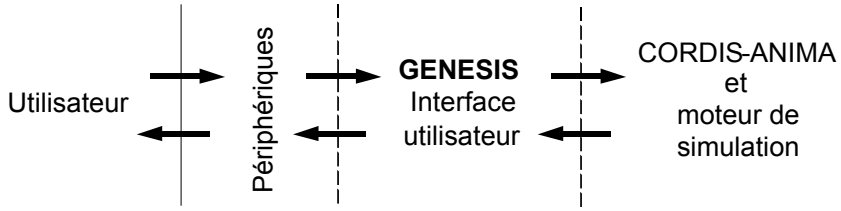


Figure 1 : l'interface GENESIS

Il est souvent difficile de distinguer les limites de l'interface utilisateur. L'interface est couramment identifiée à ce qui est visible ou manipulable (fenêtre, boutons, interpréteur de commandes, etc.), qui n'est pourtant que la partie émergente de l'iceberg. C'est pourquoi on peut substituer à la notion d'interface celle un peu plus générale de système interactif. Les méthodes, principes et théories relevant de l'Interaction Homme Machine ou de l'Ergonomie qui sont considérés dans cette partie concernent le *système interactif* GENESIS dans son ensemble.

3.1.2 - Vertus premières

Une interface doit présenter un certain nombre de qualités générales dont : l'utilité, l'efficacité, la facilité d'apprentissage, l'utilisabilité, et l'agrément.

Utilité ; transparence

La vocation d'un système interactif est de permettre la réalisation d'une tâche par la collaboration de l'homme et de la machine, c'est-à-dire d'être *utile*.

L'utilisateur n'est pas *à priori* intéressé par le système lui-même, mais par sa fonction. Un enjeu très souvent cité dans la littérature – et, au demeurant, souvent contredit par les logiciels les plus courants – est donc que l'interface n'apparaisse pas comme un obstacle entre l'utilisateur et la tâche qu'il souhaite réaliser.

Il faut « autant que possible, rendre la tâche dominante et les outils invisibles » [Norman.90]. Caelen appelle ainsi à une certaine *transparence* de l'interface. Cependant, il ajoute aussitôt que l'interface doit aussi opposer des résistances pour que l'activité soit contrôlable. Une interface totalement transparente « perdrait son rôle structurant » [Caelen.96a].

Efficacité

Un système interactif doit permettre de mener à bien efficacement la tâche (ou les tâches) pour laquelle il a été conçu. *L'efficacité* a fait l'objet d'une attention soutenue dans l'histoire de l'Interaction Homme-Machine.

Facilité d'apprentissage

Une interface doit présenter en elle-même des propriétés « auto-pédagogiques ». Aucune notice, mode d'emploi ou autres aides en ligne ne permet de compenser les défauts pédagogiques intrinsèques d'un outil. C'est donc bien dès la conception que la nécessité de l'apprentissage doit être prise en compte.

Utilisabilité

L'utilisabilité d'un système désigne, de façon très générale, sa facilité d'emploi[1]. Elle s'analyse de très nombreuses manières, tel le risque d'erreur induit par le système, la visibilité des actions possibles, la nature des *mappings* entre action et effets, etc.

Agrément

Enfin, une interface se doit de limiter l'anxiété des utilisateurs, d'être esthétique, etc. L'agrément constitue une qualité des systèmes interactifs. Comme l'explique Caelen : *«* Une interface a pour fonction de motiver rationnellement et émotionnellement un individu devant une machine. Elle est ergonomiquement adaptée si elle permet à l'usager d'agir rationnellement et efficacement, mais aussi – ne l'oublions pas – de manière agréable, de façon à soutenir la motivation d'usage *»* [Caelen.96a].

[1] Certains auteurs regroupent sous ce terme la plupart des fonctions et enjeux présentés dans ces paragraphes. La notion d'utilisabilité désigne alors à la fois « la performance de réalisation de la tâche, la satisfaction que procure l'utilisation de l'objet et la facilité avec laquelle on apprend à s'en servir » [Usabilis].

3.1.3 - Conception

La conception d'un système interactif suppose alors de considérer un certain nombre de points clés.

La tâche – les tâches

Le système doit avant tout procéder de la tâche pour laquelle il est conçu :

> « La question qu'on se pose aujourd'hui est 'Comment devrions-nous concevoir l'interface'. Je remplacerais cette question par une autre, assez différente. Je demanderais plutôt : 'Quels outils devraient être proposés pour la tâche ?' » Norman in [Rheingold.90] [1].

Considérer la tâche pour concevoir le système suppose, à minima, qu'elle soit précisément caractérisée. L'*analyse des tâches* a donc constitué une phase essentielle de notre travail.

Le noyau fonctionnel ou service

La tâche, parfois, dépend d'un noyau fonctionnel ou *service* sur lequel est construit le système interactif. Une bonne compréhension de ce système est alors nécessaire pour concevoir l'interface. Dans le cas de GENESIS, la tâche « créer » est structurée par le recours à CORDIS-ANIMA et à la simulation interactive et multisensorielle d'objets physiques.

L'homme, l'utilisateur et son environnement

Un système interactif doit être adapté à l'homme auquel il est destiné. En conséquence, comme l'explique Johnson[2], nous avons été amenés à faire des hypothèses sur les connaissances initiales de l'utilisateur de GENESIS, son expérience passée, ses facultés d'apprentissage et son environnement de travail habituel. Plus généralement, nous nous sommes appuyés sur quelques principes de la psychologie et de la cognition humaine.

L'ordinateur « communiquant », l'état de l'art en IHM

Enfin, les systèmes interactifs sont très largement conditionnés par l'ordinateur qui les supporte. La conception d'une interface ne peut être envisagée sans une bonne connaissance de l'état de l'art en matière d'Interaction Homme-Machine, et notamment :

- Des styles d'interaction *aujourd'hui* possibles avec un ordinateur ;
- Des modèles d'architectures logicielles ;
- Des outils de développement existant.

Ces différents points, résumés sur la figure 2, sont déterminants pour la conception d'une interface. Ils montrent qu'une interface doit procéder autant de l'homme que de la machine, c'est-à-dire des deux partis en présence. Nous nous appuyons sur cette dualité dans les considérations qui suivent.

---

[1] Traduction de l'auteur. « *The question posed today is in the form "How should we design the interface?" I would replace that question with a quite different one. I would ask: "What tools should be provided for the task?*».

[2] [Johnson.92 p3].

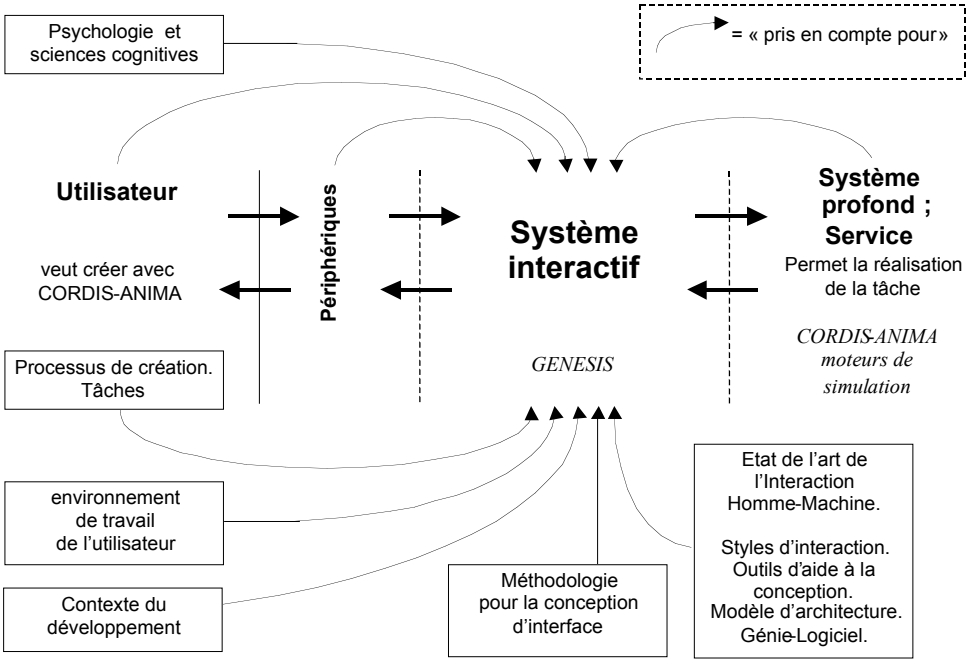


Figure 2 : facteurs en jeu pour la conception de l'interface GENESIS

3.2 - L'homme dans son environnement

Capacités perceptives et motrices, mémoire, apprentissage, capacités de réflexion et de compréhension, comportement social, peurs et rejets irrationnels…Tous les aspects du comportement humain influencent, à un moment ou à un autre, la qualité de l'interaction entre l'homme et l'ordinateur. Tous, en conséquence, méritent l'attention du concepteur d'interface – qui à minima est amené à faire des suppositions implicites les concernant.

Pour concevoir GENESIS, nous avons donc cherché à mieux connaître les bases de la psychologie et des sciences cognitives impliquées dans l'interaction homme-machine. Ce chapitre précise celles qui nous ont été les plus utiles. Il envisage successivement :

- Quelques données pratiques relatives aux capacités perceptives et motrices humaines ;
- Le modèle du *processeur humain* ; l'approche GOMS proposée par Card & al ; la théorie de l'action de Norman relative à l'interaction de l'homme avec ses outils ;
- L'approche écologique de la perception, et notamment les travaux de Gibson sur la perception visuelle ;
- Quelques notions relatives aux capacités d'apprentissage de l'Homme, et notamment les trois « mentalités » proposées par Bruner.

3.2.1 - Psychophysique et interaction homme-ordinateur : données pratiques sur l'homme percevant et agissant

Une première approche en sciences cognitives consiste à établir des lois comportementales en se basant sur ce qui est reproductible, observable et/ou mesurable du comportement humain, sans forcément que des modèles susceptibles de les expliquer ne soient recherchés. Ce paragraphe présente quelques données pratiques qui ont été ainsi obtenues[1].

[1] Une approche plus détaillée ainsi que de nombreux renvois bibliographiques pourront être trouvés dans [Baecker&al.95c], [Barnard.95], [Murch.95] par exemple.

La représentation des quantités

La psychophysique montre que la perception des phénomènes quantifiables suit très souvent, en première approximation, une loi *logarithmique*. C'est le cas par exemple pour la perception de la hauteur tonale (un doublement de fréquence correspond à la perception d'une octave), de la puissance d'un son (l'échelle des décibels correspond à cette propriété), de l'intensité d'une lumière, etc. En ce qui nous concerne la propriété logarithmique de la perception peut être exploitée par exemple pour mieux représenter certaines données quantifiables ou concevoir de meilleurs projections des actions sur leurs effets (« *mappings* »).

Limites de la perception ; perception absolue / perception relative

La psychophysique établit également les *limites* de nos capacités perceptives. On distingue notamment :

- Les limites de la plage de sensibilité – c'est-à-dire des valeurs minimales et maximales selon une échelle quelconque en deçà ou au-delà desquelles les stimuli ne provoquent plus de percept ;
- Les limites de la perception relative – c'est-à-dire la capacité du sujet à discriminer deux stimuli de même type mais légèrement différents, lorsqu'ils sont présentés au même moment ou successivement ;
- Les limites de la perception absolue – c'est-à-dire de notre capacité à estimer une quantité ou une qualité suivant une échelle donnée, lorsque le stimulus est présenté isolément et donc qu'aucune comparaison n'est possible.

De nombreuses expériences offrent des résultats opératoires quant aux limites de la plage de sensibilité, de la perception absolue et de la perception relative de divers types de stimuli. Il convient de retenir que les jugements absolus sont plus difficiles que les jugements relatifs : il est possible de discriminer des centaines d'intensités différentes de façon relative, mais seulement 5 à 12 niveaux absolus ou *catégories* sont peuvent être identifiés le long d'une dimension de la perception [Baecker&al.95e].

Quelques notes sur la perception visuelle

Du fait de la généralisation des interfaces graphiques (*Graphical User Interface*, GUI), la perception visuelle est particulièrement importante dans le contexte de l'interaction homme-ordinateur. Nous présentons ci-après quelques données utiles [Baecker & al.95e], [Hollan&al.97].

- Limites absolues

  La figure 3 synthétise quelques-unes des limites absolues de la perception visuelle.

Type de stimuli	Limite absolue	Limite pratique
Ecriture, nombres, lettres, … (signes)	26+10 (+…)	26+10 (+…)
Formes géométriques 2D (carré…)	15	5
Tailles d'une forme	5	3
Intensités d'une lumière	3	2
Couleurs (teinte)	9	9
Couleurs (combinaisons teinte/saturation/intensité)	24	9

Figure 3 : limites absolues pour la perception visuelle d'après [Baecker & al.95e].

Des combinaisons de ces différentes dimensions permettent d'augmenter le nombre de stimuli qui peuvent être discriminés de façon absolue. Ces données ont été prises en considération notamment pour la conception de la représentation des objets CORDIS-ANIMA dans GENESIS.

- Couleurs

Le développeur et l'infographiste sont habitués à manipuler certains ensembles de paramètres pour spécifier les couleurs –le système RVB (ou RGB en anglais) et le système CMJN (ou CMYK) étant sans doute les plus courants. Ces paramètres ne coïncident ni avec les paramètres physiques (ceux que mesurera le physicien : composition spectrale, intensité, etc.), ni avec les dimensions de la perception. En matière de perception de la couleur, le système de Munsell fait référence. Il distingue (d'après [Marcus.95]) :

- Le teinte (*hue*), qui correspond à la longueur d'onde *perçue*.
- La saturation (*chroma*), qui correspond à la pureté de la couleur sur une échelle allant du gris à la variante la plus pure.
- L'intensité lumineuse (*value*) qui se mesure sur une échelle allant du noir au blanc.
- L'éclat (*brightness*), qui correspond à la quantité d'énergie émise – déterminée, en général, par le réglage de l'écran.

Ces quatre dimensions sont corrélées : un changement de l'une d'entre elle peut affecter les autres.

Les données psychophysiques relatives à la perception de la couleur sont nombreuses. Nous en citerons quelques-unes (d'après [Salomon.90], [Marcus.95], [Marcus.97], [Murch.95]) :

- Notre système visuel rend difficile la perception de détails fins dans les courtes longueurs d'ondes, notamment les bleus.
- Les capacités de perception relative dépendent de la teinte dominante. La discrimination de petites variations dans les rouges ou violets est plus mauvaise que dans les jaunes, bleus et verts.
- La taille d'une surface influence la perception de sa couleur, et réciproquement. La capacité discriminatoire des couleurs (absolue comme relative) diminue lorsque la surface devient petite.
- La sensitivité aux couleurs varie entre le *focus* (le lieu ou le sujet focalise son attention) et la périphérie du champ de vision, où les rouges et les verts sont particulièrement peu perceptibles.
- La perception d'une couleur est largement conditionnée par l'environnement – notamment la lumière ambiante et les autres couleurs présentes dans le champs de vision.
- La perception ne distingue pas d'ordre entre les teintes (d'après [Salomon.90]). L'ordre du spectre est parfois présenté comme cohérent, mais ce point reste discuté.
- Il est mal indiqué d'utiliser la couleur pour communiquer des informations de nature symbolique ; les symboles et affects associés aux couleurs varient considérablement d'un individu à l'autre et surtout d'une culture à l'autre.
- Enfin, une donnée simple mais non sans conséquence pour le concepteur d'interface est qu'approximativement 8% des hommes sont atteints de formes plus ou moins prononcées de daltonisme.

- Analyse / reconnaissance

Un principe assez général en ce qui concerne la perception humaine et particulièrement valide pour la perception visuelle est la capacité de l'homme à reconnaître des *patterns* ou des structures porteurs de sens à partir des stimuli qui lui parviennent. La perception est ainsi conditionnée par l'expérience du sujet et par les hypothèses que cette expérience lui permet de faire face à ce qu'il voit. Lindsay et Norman disent ainsi que la perception visuelle est « dirigée par le concept » (*conceptually driven*, [Lindsay&Norman.77]) : nous ne percevons pas des lignes, des formes ou des surfaces séparées, mais des objets – propriété dont tirent partie les illusions optiques.

Par ailleurs, nous noterons que l'œil est particulièrement sensible au contexte dans lequel les stimuli sont présentés et que la perception est un phénomène dynamique, le *focus* navigant entre les parties différentes de la scène suivant les besoins[1].

La perception visuelle est éminemment complexe, et ce paragraphe n'en aura proposé qu'une approche très partielle. Il convient donc de considérer la mise en garde de Baecker :

> « ce qui est vu n'est souvent pas ce qui est présenté. (…) Lorsque nous organisons nos écrans, nous ne devons pas supposer automatiquement que nos intentions peuvent être aisément traduites en une présentation et une communication effective. Il nous faut travailler pour obtenir le résultat désiré » [Baecker & al.95e] [2].

Dans la partie II, nous proposerons quelques principes et guides ergonomiques inspirés de ces analyses, qui nous auront été plus directement utiles pour la conception de GENESIS.

Les temps de réaction et la loi de Hick

La loi de Hick (voir par exemple [Barnard.95]) estime le temps nécessaire à un sujet pour désigner, parmi plusieurs alternatives présentées simultanément, celle qu'il désire. Elle s'énonce ainsi :

$$T = a + b * \ln(N)$$

où a et b sont des constantes dépendant du sujet et du système de présentation, et N le nombre d'alternatives présentées.

La loi de Hick est ici donnée essentiellement à titre d'exemple : elle montre que l'approche behavioriste empirique du comportement humain peut s'appliquer à des processus cognitifs de relativement haut niveau.

Les capacités kinesthésiques et la loi de Fitts

La loi de Fitts – et ses dérivées – relie le temps nécessaire pour atteindre une cible à sa distance et à sa taille. Elle montre que les analyses psychophysiques adressent également les capacités motrices humaines.

Fitts a étudié à l'origine le temps nécessaire pour atteindre une cible placée dans un espace monodimensionnel. On trouvera dans [MacKenzie.95] une étude bibliographique des multiples développements consécutifs à cette première approche.

---

[1] Nous discuterons à nouveau de ce dernier point au paragraphe 3.2.3 - consacré à l'approche écologique de la perception visuelle de Gibson.

[2] Traduction de l'auteur. « *What is seen is very often not what is presented. (...) In structuring our screens, we must not automatically assume that our intentions can be easily translated into effective presentation and communication – we must work to achieve the desired result* ».

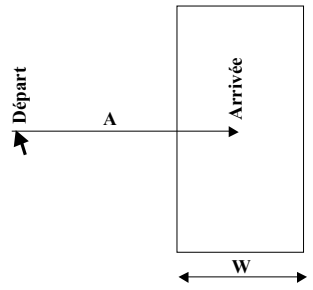


Figure 4 : la loi de Fitts

$$Temps = a + b * \ln\left(\frac{A}{W} + 1\right)$$

La formulation dite de Shannon de la loi de Fitts est considérée comme la plus efficace pour le contexte des interfaces homme-ordinateur adressées à l'aide d'une souris (figure 4 ci-contre).

De façon générale, la loi de Fitts montre que plus la cible est petite et lointaine, plus elle est difficile à trouver. Sans l'appliquer de façon quantitative, nous avons considéré ses conclusions à plusieurs reprises – par exemple pour les surfaces de « sélectionnabilité » qui entourent les représentations graphiques des modules.

3.2.2 - L'approche « computationnelle » et les modèles de l'homme face à ses outils

Modéliser le comportement humain est sans doute l'un des objectifs les plus ardus des sciences cognitives. Parmi les différentes approches, l'approche « computationnelle » est particulièrement riche dans le contexte des interfaces Homme-Ordinateur, notamment lorsqu'il s'agit de traiter les problèmes d'optimisation. Elle s'inspire des concepts et vocabulaires de l'Informatique (signaux s'entrée et de sortie, mémoire, ressources…) pour envisager l'Homme comme un processeur complexe en charge de traiter l'information qui lui parvient et d'agir en conséquence (voir [Lindsay&Norman.77], [Barnard.95], [Johnson.92 pp114-150], [Baecker.95g] par exemple).

Ressources. Les Mémoire(s)et le « nombre magique » de Miller.

Le fait de considérer l'Homme comme un processeur suggère l'existence de ressources (de perception, de calcul, de traitement, de mémorisation…) dont la capacité est limitée[1]. La *charge cognitive* d'une tâche, c'est précisément la quantité de ressources impliquée dans sa réalisation.

En ce qui concerne la mémoire, on distingue traditionnellement la *mémoire à long terme* de la *mémoire à court terme*. La mémoire à long terme est essentiellement une mémoire de stockage ; elle reposerait notamment sur l'établissement de connexions neuronales. Le rôle de la mémoire à court terme, aussi appelée *mémoire de travail* est de conserver momentanément les informations en cours de traitement. Elle sert donc notamment d'intermédiaire entre la mémoire à long terme et l'utilisation de son contenu[2] – le temps de « chargement » n'étant pas négligeable. C'est la mémoire à court terme qui est par exemple sollicitée pour lire un texte, retenir un numéro de téléphone, etc. Du point de vue neuronal, l'information y serait stockée sous la forme de potentiels électriques.

La mémoire à court terme a été assez largement étudiée. On sait qu'elle est volatile : les informations qui y sont stockées s'évanouissent en quelques secondes ou quelques dizaines de secondes [Shneiderman.98, p355]. On sait aussi que la quantité d'informations qu'elle peut contenir est limitée, et que différentes informations peuvent y être en concurrence.

[1] Les ressources de l'homme peuvent devenir « critiques » lorsque la charge cognitive augmente – par exemple lorsqu'il s'agit d'écouter deux personnes en même temps ou de retenir une suite importante de chiffres.

[2] Notons que cette idée peut être critiquée eu égard à certains sujets qui conservent une mémoire à long terme opératoire mais ont perdu toute capacité mémorielle à court terme.

Le « nombre magique » de Miller (1957), probablement l'une des caractérisations les plus célèbres des ressources du « processeur humain », indique que l'homme peut retenir jusqu'à 7(±2) éléments différents. Ces éléments ne doivent cependant pas être comptés en unités physiques (mettons, en « bit »), mais en éléments d'information ou *mnèmes* – *chunks* en anglais. Un élément, ce peut être aussi bien un simple chiffre qu'un numéro de téléphone complet, pour peu que ce dernier soit envisagé par le sujet comme une information unique. Il est donc essentiel que l'information soit structurée et présentée de façon cohérente pour qu'elle puisse être assimilée sans qu'augmente considérablement la charge cognitive de la mémoire. De ces résultats se déduisent, par exemple, les principes de *cohérence* et de *conformité* que nous présenterons au chapitre suivant.

Nous évoquerons enfin une dernière capacité mémorielle importante pour l'Interaction Homme-Machine : la mémoire du corps (*muscle memory*). La mémoire du corps est une mémoire implicite, utilisée de façon inconsciente pour réaliser toute action motrice (rouler à bicyclette, jouer d'un instrument de musique, saisir un texte au clavier, etc). Il convient de la considérer pour concevoir, par exemple les combinaisons des raccourcis clavier d'utilisation récurrente au sein d'une interface.

Le processeur humain selon Card & al

Le *modèle du processeur humain* de Card, Moran et Newell (the Model Human Processor, MHP, [Card & al.83]) constitue une contribution essentielle à l'approche « computationnelle » et a profondément marqué l'Interaction Homme Machine[1].

Le MHP est constitué de trois systèmes indépendants : sensoriel, cognitif et moteur (figure 5). Chaque système regroupe une mémoire à court terme et un processeur, caractérisés par un ensemble de paramètres (capacité et persistance de la mémoire, durée du cycle du processeur par exemple).

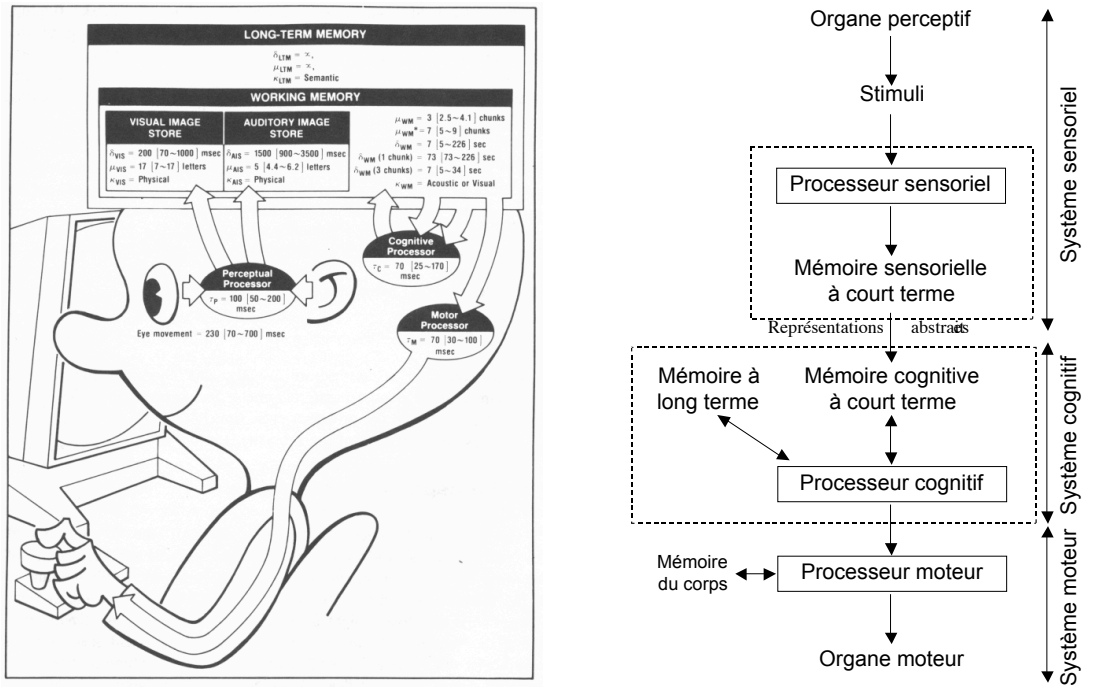


Figure 5 : le processeur humain de Card, Moran et Newell.
Gauche : le schéma originel de Card & al. Droit : un parcours d'information ([Coutaz90])

[1] Une présentation détaillée du MHP pourra être trouvée dans [Coutaz90].

Dans le MHP, un stimulus est d'abord pris en charge par le processeur sensoriel et codé dans le *buffer* sensoriel, puis lu par le processeur cognitif. Ce dernier est basé sur un cycle reconnaissance-action. Il reconnaît d'abord le contenu du *buffer* pour construire une représentation mentale de l'environnement, s'aidant notamment des *connaissances factuelles* stockées dans la mémoire à long terme. Il détermine, ensuite, l'action à entreprendre à l'aide des *connaissances procédurales* de la mémoire à long terme. L'action est alors prise en charge par le système moteur.

Les paramètres des différents sous systèmes[1], confirmés dans une certaine mesure par l'expérimentation, constituent des mesures des performances humaines. Avec le MHP, il est possible d'estimer la complexité d'une opération sensorielle, cognitive ou motrice élémentaire et le temps nécessaire à sa réalisation.

Le processeur humain de Card & al est ainsi à la fois un cadre de réflexion et un premier pas vers un modèle *quantitatif* permettant la prédiction de certains comportements humains. Il constitue un intermédiaire entre une approche théorique et appliquée de la psychologie humaine.

L'homme face à ses outils (1) : le modèle GOMS

L'approche GOMS (*Goals, Opérators, Methods and Selection rules* [Card&al.83]) effectue un pas supplémentaire dans cette direction. Elle permet de modéliser le comportement d'un utilisateur qui interagit de façon *rationnelle* avec un outil parfaitement maîtrisé (et essentiellement informatique) pour atteindre un but bien déterminé.

L'approche GOMS postule que le comportement de l'utilisateur face à l'outil peut se décrire avec quatre éléments [2] :

- Ses objectifs (*Goals*). Chaque objectif se décompose en objectifs plus simples. Par exemple, *corriger un texte* peut faire appel à l'objectif *supprimer un mot*.
- Les Opérations (*Operators*), qui sont les actes perceptifs, moteurs ou cognitifs élémentaires qu'il réalisera effectivement pour atteindre l'objectif (ou les objectifs) qu'il s'est fixé. Par exemple : *regarder à tel endroit*, *appuyer sur une touche*, etc.
- Les Méthodes (*Methods*) ou procédures, qui décrivent comment un objectif peut être atteint et représentent un savoir-faire de l'utilisateur. Une méthode est une séquence d'objectifs et d'opérations élémentaires, comprenant éventuellement des conditions (par exemple : *si je suis en bas de la page, alors remonter*).
- Les Règles de sélection (*Selections rules*), qui permettent de choisir entre plusieurs méthodes lorsqu'elles sont en conflit, c'est-à-dire lorsqu'elles permettent d'atteindre par des voies différentes un même objectif. Une règle de sélection est une déclaration conditionnelle portant sur l'état de l'outil ou des mémoires de l'utilisateur.

Une analyse GOMS consiste alors à décomposer de façon récursive un objectif en méthodes et sous-objectifs, et ce jusqu'au niveau des opérations élémentaires.

La hiérarchie complète d'opérations élémentaires finalement obtenue est conditionnée par l'outil avec lequel l'utilisateur se propose d'interagir pour atteindre son but. Dans le cas ou cet outil est informatique, elle dépend de l'interface et de son organisation.

---

[1] Dont par exemple la loi de Fitts pour le système moteur.

[2] [Coutaz.90], [Shneiderman.98 p55] et [Johnson.92 p127] exposent chacun plus en détail le modèle GOMS.

Les opérations élémentaires considérées pour l'analyse déterminent, elles, sa profondeur. On peut descendre ainsi jusqu'au niveau de la frappe des touches (*keystroke level*), c'est-à-dire jusqu'aux opérations perceptives, cognitives et motrices que peut prendre en charge le processeur humain MHP. Dès lors, il devient possible d'estimer la difficulté d'une tâche quelconque et le temps nécessaire pour la réaliser. GOMS autorise ainsi en théorie une évaluation *quantitative* de *l'efficacité* d'un système interactif.

L'approche GOMS reste néanmoins limitée ([Johnson.92], [Barnard.95]). Elle suppose notamment que l'utilisateur dispose à l'avance du savoir nécessaire à la réalisation de la tâche et ne permet de prendre en compte ni les difficultés d'apprentissage, ni les cas d'erreurs, ni les cas où une réflexion doit être mise en œuvre pour résoudre un problème. Elle ne concerne, en fait, que les utilisateurs experts et les tâches réalisées dans une interaction concrète avec l'outil. Enfin, elle est orientée vers l'évaluation d'un système existant (ou entièrement spécifié) et non pas vers l'aide à la conception.

Aussi, en ce qui nous concerne, l'approche GOMS a été plus utile du point de vue théorique que pratique. Nous n'aurons donc pas cherché à la mettre en œuvre directement. En revanche, nous nous sommes inspirés du principe d'analyse récursive des objectifs qu'elle propose (voir le chapitre 5 consacré à la méthodologie). Elle nous a ainsi offert quelques éléments de réflexion, invitant notamment à ce que tout choix ergonomique soit analysé sous l'angle des opérations élémentaires qu'il impose, et ce le cas échéant jusqu'au niveau de la frappe des touches.

L'homme face à ses outils (2) : La « théorie de l'action » de Norman

Plusieurs autres modèles de l'homme utilisant un outil existent dans la continuité de l'approche GOMS[1]. Tous, cependant, appellent des critiques similaires – on parle en la matière de « recherches préliminaires ». Plutôt que de chercher un modèle quantitatif, il est intéressant d'envisager une théorie générale. En la matière, les *sept étapes de l'action* de Norman (aussi connues sous le nom de *théorie de l'action*) sont d'un précieux recours pour le concepteur d'interface [Norman.88].

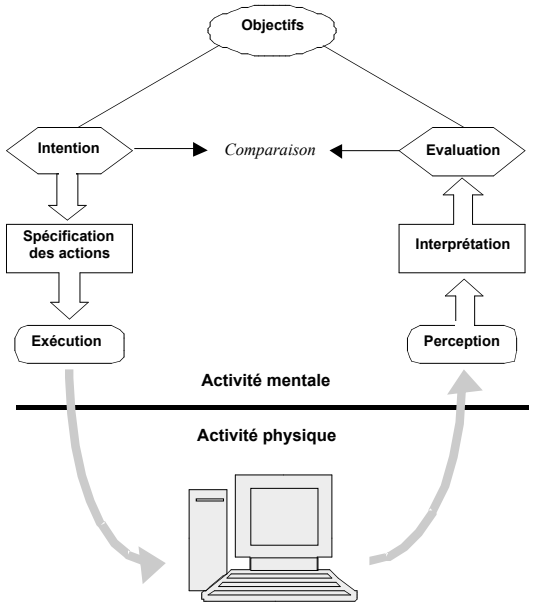


Figure 6 : les* 7 étapes de l'action *d'après [Norman.88].

Dans le modèle de Norman, l'objectif de l'utilisateur joue à nouveau un rôle pivot et conditionne son comportement. La première des 7 étapes de l'action est ainsi :

- établir un but ;

Trois étapes permettent la mise en œuvre d'une action qui rapproche du but :

- former une intention ;
- spécifier une suite d'actions ;
- exécuter les actions ;

Trois étapes symétriques permettent de prendre connaissance de l'effet des actions :

- percevoir l'état du système (sensation) ;
- interpréter l'état du système (cognition) ;
- évaluer l'état du système par rapport à l'intention et au but originel.

Le cycle action/évaluation peut ensuite recommencer.

[1] Par exemple le *Keystroke Level Model* (KLM) par Card, Moran et Newell ou le *Natural GOMS Language* (NGOMSL) par Kieras. Voir [Johnson.92 pp114 – 150] ou [Baecker&al.95h].

Les *7 étapes de l'action* de Norman rendent compte, par exemple, des problèmes qui se posent lorsque le système ne permet pas la réalisation des intentions de l'utilisateur (la spécification des actions est impossible), lorsque l'évaluation de l'état du système est difficile à effectuer ou lorsque le système trahit les suppositions de l'utilisateur quant au résultat de ses actions. Elles constituent un cadre théorique pour l'analyse du comportement des utilisateurs face à une interface. Elles sont en outre à l'origine de plusieurs des principes que nous présenterons au chapitre suivant.

3.2.3 - Action, perception, représentation mentale : l'approche écologique de Gibson

L'approche *computationnelle* stipule que sensation, perception, cognition et contrôle moteur correspondent à des processus distincts. Pour étudier l'un de ces processus, il convient d'éliminer toutes les perturbations que peuvent causer les autres[1]. Les études séparées visent d'abord à établir les lois qui régissent les différents processus, puis, à terme, des modèles plus généraux du comportement humain.

La *théorie écologique de la perception*, initiée par Gibson, stipule à l'inverse qu'action et perception sont indissociables et notamment que la perception est un processus *actif* [Gibson.79]. Gibson soutient que la perception visuelle est basée sur une analyse dynamique de *flux* et en particulier des perturbations du *réseau optique ambiant* plutôt que sur celle des formes qui impressionnent l'image rétinienne. Ces flux renseignent l'homme à la fois sur l'environnement et sur lui-même.

Pour Gibson, par exemple, la perception de l'espace ne résulte pas de la vision stéréoscopique. Elle est due à la capacité de l'homme à se mouvoir dans l'environnement et aux modifications du réseau optique consécutives aux mouvements, notamment le masquage progressif de certaines formes par celles qui sont plus proches. De même, la représentation mentale d'un objet se construit de façon active, au fur et à mesure que le sujet s'en approche, le contourne, l'envisage sous plusieurs angles.

L'analyse de Gibson porte essentiellement sur la perception visuelle. Elle est donc particulièrement utile pour l'Interaction Homme Machine, alors même qu'aujourd'hui les interfaces graphiques utilisateur (Graphical User Interface, GUI) sont les principales voies d'interaction avec l'ordinateur[2]. L'approche écologique est cependant généralisable. Nous dirons que la représentation mentale de l'environnement se construit de manière active, et qui plus est par une collaboration de *l'ensemble* des moyens de perception et d'action : on agit tout en percevant et on perçoit tout en agissant[3].

Plusieurs résultats pragmatiques peuvent être retenus de l'approche écologique de la perception. Dans GENESIS, par exemple, nous avons cherché à guider l'attention de l'utilisateur, à induire un parcours entre les différentes informations portées à l'écran. Nous avons par ailleurs conçus de front l'affichage et les actions possibles afin que les deux soient cohérents et assuré que l'interface rende compte dynamiquement des actions entreprises par l'utilisateur.

---

[1] Par exemple, pour étudier la perception visuelle sous l'angle de la psychophysique, on tente d'interdire le mouvement de la tête, voire de la rétine – ou on présente des images statiques sur des durées très courtes.

[2] Gibson a introduit en outre le concept d'affordance (offrande), repris plus tard par Norman, qui est particulièrement important dans le contexte de l'Interaction Homme-Machine. Nous le détaillerons au chapitre II.

[3] Cela s'applique parfaitement à la perception de l'espace par le canal audio, qui est éminemment dynamique (mouvement relatif des sources sonores et de l'auditeur, mouvement de la tête…). Par ailleurs, l'approche de l'ACROE en matière de geste se place dans la continuité de l'analyse de Gibson, stipulant qu'il est réducteur d'envisager le canal gestuel comme une *commande*, alors qu'il est éminemment interactif et mêle lui aussi action et perception.

3.2.4 - Action, représentation, symboles : les modes de la pensée selon Bruner

Nous évoquerons, pour terminer, les travaux des psychologues Jean Piaget et Jerome Bruner quant à l'apprentissage. Ces travaux ont contribué, dans les années 70, à l'émergence des principaux concepts des interfaces aujourd'hui répandues[1] et apportent aujourd'hui encore un éclairage utile sur la « psychologie de l'utilisateur ».

Selon PIAGET, le développement cognitif, de l'enfant à l'âge adulte, traverse plusieurs étapes : sensori-motrice, pré-opératoire, opérationnelle concrète, et opérationnelle formelle – le terme opérationnel renvoyant à la capacité de se représenter intérieurement une action et d'effectuer les opérations mentales visant à chercher des données sur le monde. En d'autres termes, l'enfant en bas âge est limité à une approche kinesthésique ou sensori-motrice du monde. Plus tard, il commence à opérer des représentations, notamment grâce à la vision, et enfin seulement se développe l'intelligence formelle et la capacité de pensée abstraite.

Bruner considère que les différentes phases du développement observées par Piaget perdurent et coexistent chez l'adulte. Elles sont alors essentielles à l'apprentissage : on peut apprendre par l'action (en faisant) ; par la vision (en voyant et voyant-faire) ; ou au niveau symbolique – la connaissance étant acquise de façon d'autant plus profonde que ces trois voies sont utilisées tour à tour. La compréhension des capacités d'apprentissage de l'homme est un problème clé pour l'Interaction Homme Machine, et l'approche de Bruner fait en la matière référence[2].

Bruner va cependant plus loin, considérant que notre esprit adopte, suivant les situations auxquelles il est confronté, différents modes de pensée ou « mentalités » bien délimités (disons : discrets). Il identifie et étudie notamment les modes *enactif* (disons : concret), *iconique* (ou visuel), et *symbolique* (ou abstrait)[3].

Ces modes de pensée s'excluent mutuellement tout en étant complémentaires. Chaque mode, en effet, a des caractéristiques et aptitudes particulières, des champs d'activité privilégiés, permet la réalisation de certaines tâche et l'acquisition et la manipulation de certaines catégories de connaissances. Aucun d'entre eux ne peut couvrir l'ensemble des problèmes auxquels l'homme se trouve confronté. Enfin, il apparaît que nous avons tous un (ou des) mode(s) de pensée privilégié(s).

Kay[4] a montré que l'ordinateur peut adresser indifféremment chacun des modes de pensée identifiés par Bruner [Kay.90] : à chacun d'eux correspond un type d'interaction avec la machine (figure 7).

---

[1] Selon Alan Kay, parfois considéré comme « le père de l'ordinateur personnel » et l'un des principaux concepteurs des interfaces WIMP (*Windows*, *Icons*, *Menus*, *Pointing*) aux laboratoires Xerox à la fin des années 70, [Kay.90].

[2] Notons que Johnson donne de nombreuses autres informations quant au processus d'apprentissage humain [Johnson.92 pp11-67]. Quelques-uns des modèles relatifs à la construction, au stockage, à la recherche et à la réactivation des informations par l'homme sont présentés dans son ouvrage, ainsi qu'un ensemble de données précises en ce qui concerne l'acquisition des savoir-faire procéduraux (*skills*).

[3] Sans ignorer pour autant l'existence d'autres modes (par exemple l'affect).

[4] Observant l'importance du mode de pensée iconique ou visuel pour les interfaces les plus courantes, Kay l'a étudié attentivement. Il rappelle notamment qu'il est toujours plus facile de construire un diagramme que de le lire [Kay.90] – point particulièrement important dans le cadre de GENESIS.

Mode	Principaux types de travaux	Situation typique dans le cas de l'interaction homme-ordinateur
Enactif ou concret	Savoir ou on est, manipuler	Manipulation de la souris, clavier… Utilisation de la mémoire du corps
Iconique ou visuel	Reconnaître, comparer, configurer. Concret.	Fenêtres, icônes, *widgets*. Manipulation directe (voir ci dessous).
Symbolique ou abstrait	Enchaîner des raisonnements complexes, développer une pensée. Abstrait.	Utilisation des langages informatiques (compilés ou en ligne de commande).

Figure 7 : les trois modes de la pensée dans le cas de l'interaction avec l'ordinateur – d'après [Kay.90]

Selon Kay, l'un des paris qui est à la base des interfaces WIMP et de la manipulation directe [1] est que les modes concrets et visuels (qui sont les premiers que nous acquérons et, disons, les plus intuitifs) permettent de parvenir à une compréhension symbolique des choses [Kay.90]. Le slogan « *doing with images makes Symbols* » (*agir* avec des *images* produit des *symboles*) célèbre dans les années 70 en est une manifestation.

Ce pari, cependant, n'est encore aujourd'hui que très partiellement réussi : les interfaces WIMP adressent essentiellement les mentalités concrètes et visuelles, laissant en grande partie de côté le mode de pensée symbolique. C'est qu'il est difficile d'élaborer une pensée symbolique face à des icônes et que réciproquement, dans une certaine mesure, « il ne peut y avoir de manipulation directe quand il s'agit de symbole » [Torvik.01]. La portée relativement limitée des environnements de programmation graphique en est une illustration.

De ce point de vue, le paradigme WIMP semble voué à un certain échec. Sans aller aussi loin, l'analyse qui précède montre qu'il n'est pas avisé de systématiquement chercher à construire une interface autour du paradigme de la manipulation directe. A la fin de la partie VII, elle nous conduira à souhaiter qu'une représentation symbolique des modèles CORDIS-ANIMA soit élaborée en complément de la représentation graphique que propose GENESIS 1.5, et à donner quelques pistes en ce sens.

[1] WIMP: Windows, Icon, Mouse and Pull-down menus - voir page suivante. Manipulation directe : voir page79. Les interfaces WIMP à manipulation directe sont aujourd'hui de loin, les plus répandues

3.3 - L'ordinateur « communiquant »

Les conditions de l'interaction homme-ordinateur ont considérablement évolué au cours des 40 dernières années[1]. Cependant, les ordinateurs sont encore aujourd'hui loin d'être malléables. Ils privilégient, en quelque sorte par construction, certains modes d'interaction. De la même manière que l'architecte se base sur les propriétés des matériaux et le savoir-faire des constructeurs pour édifier des bâtiments, il nous faut pour concevoir GENESIS connaître l'état de l'art de l'Interaction Homme-Ordinateur. C'est l'objet de ce paragraphe, qui présente :

- Les styles d'interaction courants et les concepts correspondants ;
- Les recherches récentes en matière d'interfaces ;
- Les environnements informatiques adaptés au développement des interfaces.
- Les bases théoriques de l'architecture logicielle des systèmes interactifs ;

3.3.1 - Interfaces graphiques WIMP

Le paradigme des interfaces WIMP (*Windows, Icons, Mouse and Pull down Menus*) est aujourd'hui presque universellement répandu[2]. Il est supporté par la quasi-totalité des systèmes d'exploitation – et la totalité des systèmes pour ordinateurs personnels.

Les interfaces WIMP se caractérisent en trois points :

- Elles reposent sur trois périphériques : la souris, le clavier et le moniteur graphique.

Vinet prend bien soin de décrire les conditions de leur utilisation : « Le lien entre la commande effectuée par l'intermédiaire de la souris et l'affichage graphique est assuré par la matérialisation d'un curseur de forme programmable, dont la position à l'écran est calculée à partir des déplacements relatifs de la souris. Le clavier est utilisé comme accès complémentaire pour la saisie de données alphanumériques, le déclenchement de fonctions particulières ou le déplacement du curseur, au moyen des touches de navigation » [Vinet.99a]. Trois actions sont réalisables à la souris : *pointer*, *cliquer*, et *tirer*.

- Elles font une utilisation intensive des fenêtres.

L'écran est subdivisé en plusieurs fenêtres superposables. A une fenêtre correspond soit une application soit un ensemble de fonctionnalités d'une application.

- Elles font appel à trois styles d'interaction bien délimités[3].

Ces styles d'interaction, essentiels en ce qu'ils sont à la base de GENESIS, sont présentés dans le paragraphe suivant.

---

[1] Un historique de l'interaction homme-ordinateur depuis les années 60 est présenté dans [Baecker&al.95b] et [Baecker&al.95a]. Parmi les étapes les plus marquantes, notons l'apparition du *temps partagé*, le développement des moniteurs graphiques (système *Sketchpad*, Sutherland, 1963) et l'apparition dans les années 80 de l'ordinateur personnel et des principes des interfaces graphiques qu'on connaît aujourd'hui.

[2] Son origine se trouve dans les recherches menées au Paolo Alto Research Center (PARC) de Xerox dans les années 70. Le système Star de Xerox est, au début des années 80, la première station de travail à interface graphique. Il est suivi en 84 du Macintosh d'Apple. Ces deux systèmes sont les principaux pionniers de l'ordinateur personnel et des interfaces WIMP.

[3] Notons que Shneiderman en relève 5 : la manipulation directe, les menus, les formulaires, les langages de commande et le langage naturel [Shneiderman.98 p73]. La classification en trois points ici proposée nous semble plus appropriée.

3.3.2 - Styles d'interaction courants : *Widgets*, manipulation directe, interaction par langage

Interaction par objets graphiques et évènements ; Widgets

Ce premier style d'interaction repose sur des objets graphiques activables ou *widget* : boutons simples, *radio butons*, menus, barre de navigation, champs de saisie, etc[1]. L'implémentation de ces différents *widgets* (et plus généralement des interfaces WIMP) est facilitée par l'existence d'outils de développement d'interfaces[2].

Manipulation directe ; notion d'objet d'intérêt

Le terme *manipulation directe* a été introduit par Shneiderman en 1983 [Shneiderman.83] pour qualifier les styles d'interaction qui apparaissaient alors. Les principes avancés par Shneiderman sont les suivants :

- *Principe 1* : représentation constante de l'objet d'intérêt (c'est-à-dire l'ensemble de données que veut éditer l'utilisateur, par exemple le document à rédiger ou l'image à retoucher).
- *Principe 2* : actions physiques directes sur l'objet d'intérêt, au lieu d'une syntaxe complexe.
- *Principe 3* : opérations rapides, incrémentales, réversibles, dont l'impact sur l'objet d'intérêt est immédiatement visible.

En résumé, on parle donc de manipulation directe lorsque l'objet d'intérêt est représenté et manipulable « comme s'il s'agissait d'un objet réel ».

La *métaphore* du bureau et les éditeurs de textes sont deux exemples typiques d'interfaces à manipulation directe. Dans le cas des éditeurs de texte, qui plus est, ce qui est affiché correspond à ce qui sera obtenu lors d'une impression. Ce dernier point, incidemment, est à l'origine du slogan « WYSIWYG » (*What You See Is What You Get*, ce que vous voyez est ce que vous obtiendrez) qui s'est développé parallèlement aux systèmes WIMP.

Selon Shneiderman la manipulation directe présente des avantages importants [Shneiderman.83] :

- Elle est utilisable avec un minimum de connaissance ; elle facilite l'apprentissage, l'auto-apprentissage, la mémorisation, et de façon générale l'utilisation par des utilisateurs occasionnels ou novices.

---

[1] A chaque *widget* correspond d'une part une apparence graphique, d'autre part un ensemble d'actions possibles, à réaliser essentiellement avec la souris.

[2] On distingue trois types d'outils d'aide au développement ([Baecker&al.95d], [Shneiderman.98 p155]. (1) Les *boîtes à outils* (*toolkit*) proposent une bibliothèque de *widgets* et un système de gestion des évènements (typiquement, le *toolkit* définit un ensemble de *callbacks*, procédures appelées lorsque un *widget* est activé par l'utilisateur). (2) Les *librairies graphiques* sont utiles pour toutes les fonctionnalités exigeantes en termes graphiques, par exemple lorsqu'il s'agit de représenter les objets d'intérêts dans les styles à manipulation directe (voir ci-dessous). (3) Enfin, des environnements plus complets (squelette d'interface, *User Interface Management Systems* UIMS, etc.) sont censés supporter l'ensemble du cycle de développement (des spécifications initiales aux tests). Ils tendent tous à imposer un modèle d'architecture logicielle spécifique et, plus important, une méthodologie de conception. Ils ne nous sont pas apparus judicieux pour GENESIS : nous avons préféré conserver une plus grande liberté dans notre travail et utiliser plus simplement une boîte à outil et une librairie de dessin, un environnement standard de développement et une bonne connaissance des principaux modèles d'architecture logicielle. L'environnement de développement utilisé pour GENESIS est précisé dans la partie V.

- Elle augmente la satisfaction des utilisateurs et diminue l'anxiété.
- Elle permet une meilleure performance des utilisateurs experts – ce point étant encore aujourd'hui discuté[1].

Par ailleurs :

- La manipulation directe tend à une immersion plus profonde de l'utilisateur au sein même du système, et en conséquence induit un engagement plus important – là où les interfaces basées sur un langage laissent entendre qu'un intermédiaire caché traduit l'intention de l'utilisateur sur l'objet qu'il souhaite éditer [Frolich.97].
- Enfin, de façon corollaire, elle tend à focaliser l'attention de l'utilisateur sur le document faisant ainsi passer l'interface au second plan au profit de la tâche à réaliser.

La manipulation directe présente cependant plusieurs faiblesses ([Shneiderman.83], [Shneiderman.98 pp204-205], [Frohlich.97], [Hutchins&al.86] [Baecker&al.95b]) :

- Elle suppose, d'abord, qu'une représentation efficace et compréhensible de l'objet d'intérêt soit possible. Baser une interface sur une représentation malvenue amène de nombreux problèmes. De même, des difficultés émergent lorsque la représentation laisse croire à tort que certaines actions sont possibles – voir aussi la notion d'offrande (*affordance*) au paragraphe 4.2.2 - ci-dessous.
- Elle ne permet pas d'effectuer de façon automatisée des actions répétitives.
- Avec la manipulation directe, il est difficile d'indiquer avec précision et de (re)trouver un objet parmi de nombreux autres qui lui sont similaires.
- La facilitation de l'apprentissage et de la mémorisation qu'on lui prête traditionnellement est en fait conditionnée par les modes de pensée privilégiés de l'utilisateur. Elle s'accorde mal avec une pensée symbolique.
- Enfin, selon Hutchins & al et contrairement à ce que considère Shneiderman, elle tend à être moins efficace pour les utilisateurs experts qu'une interface basée sur un langage.

Nous avons déjà discuté de l'importance du raccourci qui transforme pour l'utilisateur le *modèle* CORDIS-ANIMA en *objet physique*, et qui permet à la *pensée physique* de se développer (voir la partie I). Compte tenu de ses avantages, la manipulation directe d'un objet CORDIS-ANIMA *représenté* s'impose naturellement comme une solution optimale pour soutenir ce raccourci, et ce malgré les critiques précédentes. Nous aurons donc décidé d'y avoir recours au sein de GENESIS.

Interaction par langages

Le *faire* et le *(perce)voir* qui sont à la base des interfaces WIMP ne sont pas adaptés à toutes les situations d'interaction homme-machine. Le recours à un langage est plus indiqués dans les domaines d'activité formelle. Les langages permettent de tirer parti au mieux des possibilités de l'ordinateur quant à la manipulation de concepts.

Quelques autres avantages plus pragmatiques peuvent en outre être cités pour le style d'interaction reposant sur un langage :

- Il permet d'exprimer de façon condensée des opérations complexes, de spécifier des enchaînements d'opérations et d'effectuer aisément des actions répétitives.

[1] Shneiderman nuance son opinion première dans un texte ultérieur, indiquant qu'elle « peut être rapide pour des utilisateurs fréquents » si elle est conçue avec attention [Shneiderman.98 p73]. Voir aussi les critiques de Hutchins & al ci dessous.

- Il permet d'indiquer avec précision les actions à effectuer ou l'objet à rechercher.
- Il permet de définir des variables, des classes, des abstractions et d'organiser ainsi plus facilement les données.
- Lorsque la syntaxe est acquise – ce qui ne va pas toujours sans difficultés – il peut être particulièrement efficace pour un utilisateur expert.

Dans le contexte des interfaces WIMP, les interactions en lignes de commande ou plus généralement via un langage restent utiles et, finalement, relativement courantes. L'utilisation de langages compilés est de loin le cas le plus fréquent pour la programmation, mais de nombreux autres exemples existent : l'édition HTML, certains éditeurs de texte ou d'image, l'interrogation de base de données, etc.

3.3.3 - Quelques innovations et tendances récentes en matière d'interaction homme-ordinateur

Si les styles d'interaction évoqués jusqu'ici sont aujourd'hui les plus courants, ils ne constituent pas un aboutissement. Ce paragraphe évoque quelques innovations et tendances récentes en matière d'interaction homme-ordinateur. En dépit d'un évident conservatisme industriel, (tant en ce qui concerne l'architecture matérielle et les périphériques qu'en ce qui concerne les principes logiciels), certaines d'entre elles se diffusent progressivement auprès des utilisateurs.

Reconnaissance de la parole

Le langage parlé est à la fois naturel et puissant. Les avantages que présenterait une interaction vocale sont donc assez évidents [Néel.96], [Caelen.96b]. Les difficultés inhérentes à sa mise en œuvre le sont cependant tout autant. Elle suppose que l'ordinateur :

- Puisse *reconnaître* des mots, des phrases, des intonations aussi, alors même que le « signal parole » est variable d'un individu à l'autre et en fonction du contexte (fatigue, stress…) ou de l'environnement (bruit, locuteurs superposés, etc.)
- Sache extraire du signal sonore une représentation sémantique, alors même que les tournures utilisées sont largement ambiguës et variables.

Langage naturel ; convivialité

Pour qu'une interaction orale soit possible, il faut encore que l'ordinateur soit à même de réagir convenablement aux sollicitations exprimées dans un langage non contraint, disons : dans la langue de tous les jours.

C'est l'objet des recherches sur les *interfaces en langage naturel*. Basées plus simplement sur une interaction au clavier, elles visent à définir des *agents interactifs* susceptibles de converser avec l'utilisateur et de médiatiser ses demandes sur l'objet d'intérêt [Cavazza.96], [Sadek.96], [Baecker&al.95]. Leur principal champ d'application concerne, à ce jour, l'interrogation des bases de données.

Multisensorialité

Dans les interfaces WIMP classiques, une partie très restreinte de nos capacités sensori-motrices est mise à contribution. La recherche d'une interaction multisensorielle, par exemple utilisant le son, constitue une tendance particulièrement importante aujourd'hui [Conversy&Beaudouin.96]. Notons que les travaux de l'ACROE en matière d'interaction multisensorielle, et particulièrement d'interaction gestuelle avec retour d'effort, participent du même mouvement. Nous expliquons dans la partie IV, chapitre 11

que nous avons décidé de ne pas les intégrer à l'ensemble du logiciel GENESIS, réservant l'interaction multisensorielle et particulièrement l'interaction gestuelle via le TGR© au mode Jeu encore en gestation.

Reconnaissance de gestes

Dans les interfaces WIMP classiques, souris et claviers utilisent un répertoire de gestes limité : le geste est essentiellement pris en compte sous la forme de changements d'états discrets (dedans, dehors, appuyé, relâché, cliquer, activer, etc.). Les recherches sur la reconnaissance de geste visent à enrichir la sémantique des actions gestuelles de l'utilisateur. Elles reposent sur une analyse étendue du « signal gestuel » par le programme ; la trajectoire, le mouvement effectué, la dynamique du geste peuvent, par exemple, être pris en compte pour inférer l'opération que voulait réaliser l'utilisateur. Nous retiendrons, en matière d'exemple, trois applications possibles des possibilités [Beaudouin.99] et [Baudel.96].

- Gestes symboliques pour la désignation des actions

Ce type d'interaction repose sur un vocabulaire de gestes.

A chaque geste (forme, mais aussi dynamique le cas échéant) correspond une commande, et éventuellement ses paramètres (figure 8).

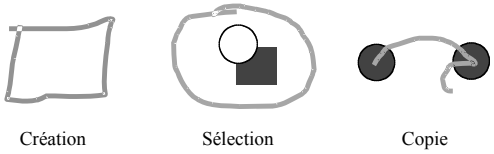


Figure 8 : reconnaissance de gestes
d'après [Beaudouin.99]

- Reconnaissance de forme et *lignes claires*

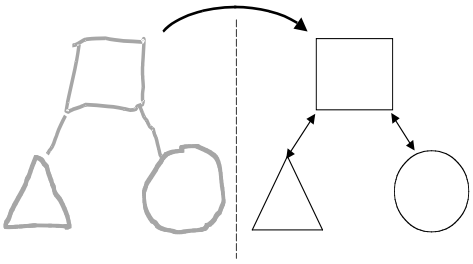


Figure 9 : reconnaissance de forme ;
d'après [Baudel.96]

Le geste effectué à la souris est peu précis. La reconnaissance de gestes permet de déduire d'une forme approximative tracée à la souris la forme géométrique que souhaitait dessiner l'utilisateur. Le geste sert alors à la fois à spécifier la commande et ses paramètres (voir [Beaudouin.99] par exemple).

Les *lignes claires* consistent, elles, à corriger un tracé en le surchargeant par étapes successives ; elles évitent qu'une courbe ne soit définie par des points de contrôles interpolés, et qu'un outil spécifique ne soit sélectionné pour les modifier [Baudel.96].

- Les *menus pointés* (*marking menus*)

Les *menus pointés* constituent une alternative aux menus traditionnels : les entrées d'un menu sont disposées en arbre. Pour sélectionner une entrée, l'utilisateur parcourt (approximativement) les nœuds de l'arbre. S'il est peu expérimenté, une pause légère dans le parcours a pour effet d'afficher les différents choix possibles au nœud considéré. S'il est expert, il peut désigner l'entrée qui l'intéresse sans qu'un retour visuel ne vienne alourdir la procédure.

Enfin, il convient de noter :

- D'une part que les recherches sur la reconnaissance de gestes sont relativement avancées. Des librairies dédiés existent (HyperStrokes par exemple) ainsi que des algorithmes de reconnaissance efficaces (en la matière, l'algorithme de Rubine fait référence [Rubine.91]). Plusieurs logiciels commerciaux les utilisent d'ores et déjà.
- D'autre part que leur mise en œuvre ne nécessite pas forcément de reconfiguration profonde de la machine : elle peut reposer sur l'utilisation des périphériques usuels que sont la souris et le clavier et compléter à moindre frais une interface à manipulation directe plus traditionnelle.

Des diverses innovations et tendances présentées dans ce paragraphe, la reconnaissance de gestes est, en conséquence, la seule que nous aurons considérée pour le développement de GENESIS (partie V, chapitre 13).

Interaction bimanuelle ; tool glass et magic lens

Une seule main est utile à la manipulation de la souris, alors que dans la vie courante nous sommes tous habitués à utiliser nos deux mains en collaboration.

L'utilisation des deux mains est asymétrique. Il semble que la main non dominante agisse avant la main dominante : elle définit le cadre d'action et est chargée des actions les plus grossières, la main dominante réalisant les plus fines (d'après Guiard, in [Baecker&al.95f]. Les interfaces *see-through* (voir à travers) utilisent ces propriétés.

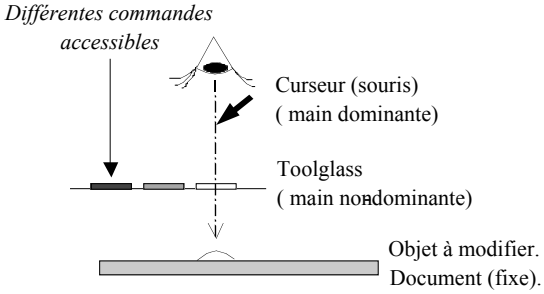


Figure 10 : Le principe du tool glass. [Bier&al.95]

Elles reposent sur deux nouveaux *widgets*, le *tool glass* et le *magic lens* [Bier&al.95], [Bier&al.93]. Le *trackball*, par ailleurs, est un périphérique spécifiquement conçu pour la main non-dominante.

Le *tool glass* (figure 10) peut être envisagé comme une grille sur laquelle figurent divers contrôles ou commandes. La main non dominante contrôle la position de la grille relativement à l'objet d'intérêt. La main dominante sert à désigner l'action qui doit effectivement être déclenchée, puis à la réaliser.

Le *magic lens* relève du même principe mais fonctionne à la manière d'une « loupe magique ». Déplacé sur une partie du document, il permet d'en faire ressortir certaines propriétés (zoom simple, filtre, affichage d'information, etc.).

Multimodalité

Nous évoquerons enfin les interfaces *multimodales*, qui font cohabiter plusieurs styles d'interaction. Une telle cohabitation pose d'importants problèmes. L'un d'entre eux est que chaque style d'interaction tend à induire un certain type d'architecture logicielle et de représentation des données des données en mémoire.

[Dujardin&Vigouroux.96] donne quelques exemples d'interfaces multimodales et principes pour leur mise en œuvre. Dans la mesure où les différents styles d'interaction présentent chacun des intérêts spécifiques, la multimodalité est une piste prometteuse. L'utilisateur, en effet, est alors à même de choisir celui qu'il préfère ou qui lui semble le plus approprié dans le contexte de sa tâche. Dans la partie V, nous verrons qu'il serait ainsi très utile qu'un langage puisse être utilisé dans GENESIS en parallèle avec la représentation graphique et la manipulation directe.

3.3.4 - Architecture logicielle des systèmes interactifs ; modèles

Les modèles d'architecture logicielle sont les résultats des recherches dans le domaine de l'architecture conceptuelle ([Coutaz.96], [Coutaz.02]). Ce sont des structures génériques qui constituent une base théorique pour le développement des systèmes interactifs. Ils définissent différentes couches idéales dans les programmes ainsi que la façon dont les fonctions et fonctionnalités s'organisent sur ces parties discrètes. Ils apportent ainsi une aide lorsqu'il s'agit de concevoir l'architecture *effective* d'un système donné. Ils visent à faciliter par exemple la maintenance et la réutilisabilité. Ce paragraphe présente d'abord quelques enjeux généraux de l'architecture des systèmes interactifs puis aborde successivement les modèles SEEHEIM, Arch, MVS et PAC[1].

Principes généraux de l'architecture conceptuelle

Plusieurs principes sont communs aux différents modèles d'architecture.

- Modularité

En architecture logicielle, un module est défini par un ensemble de données et fonctionnalités internes, et par un ensemble de *services* qu'il peut fournir. Idéalement, la structure interne d'un module (par exemple sa structure de données ou encore son apparence à l'écran) peut être remplacée sans que le reste du code n'ait à être modifié. La nature des modules n'est pas imposée par le principe de modularité, et dépend largement des structures effectivement mises en œuvre lors du développement. Les modules peuvent être suivant les cas des librairies, des classes, des modules (au sens du langage C, i.e : des fichiers), des processus indépendants, des *agents* (au sens des paragraphes suivants), etc.

- Séparation du noyau fonctionnel et de l'interface

Le second principe constitue, en fait, un premier niveau de modularité. Il consiste à ce que le *noyau fonctionnel* et *l'interface* soient séparés.

Le noyau fonctionnel (parfois appelé domaine du logiciel) regroupe les structures de données fondamentales représentant les objets édités et les actions possibles sur ces données indépendamment de toute interaction effective avec l'utilisateur. Il constitue un premier niveau de représentation des fonctionnalités profondes du logiciel. L'interface apparaît alors comme la couche logicielle en charge de l'interaction avec l'utilisateur. Son rôle est de médiatiser auprès de ce dernier l'état des données du noyau fonctionnel et d'organiser l'appel aux services en fonction de ses actions. Comme précédemment, la communication entre noyau fonctionnel et interface peut prendre des formes diverses suivant les nécessités et le système sur lequel le logiciel est développé (appel de procédures ou méthodes, messages, mémoire partagée, *pipe*, etc.).

Seeheim et Arch : les fondations

Les modèles Seeheim et Arch sont les premières architectures conceptuelles qui ont été proposées. Une architecture Seeheim (Pfaff, Seeheim Workshop, 1985) fait apparaître quatre composants essentiels (voir figure 11) :

- L'Application ou noyau fonctionnel, qui définit un ensemble de services de base.
- Un composant d'Interface qui redéfinit les services et structures de données de cette dernière pour les adapter aux nécessités de leur utilisation. Il assure que l'Application ne connaît pas le reste de système.

[1] Précisons toutefois que d'autres modèles existent, tels PAC-Amadéus.

- Un composant de Présentation qui spécifie le comportement et l'organisation du système pour l'utilisateur.
- Un Contrôleur de dialogue, médiateur entre la Présentation et l'Interface de l'Application.

L'interface utilisateur est constituée de ces trois derniers composants.

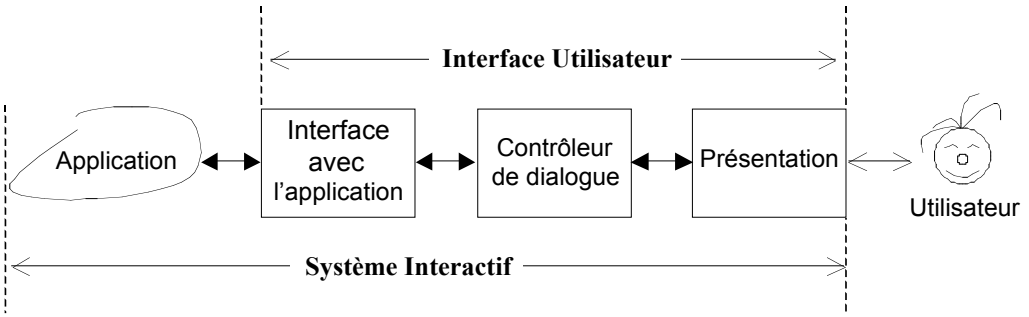


Figure 11 : le modèle Seeheim

Le modèle Arch est un développement de Seeheim plus adapté au cas des interfaces graphiques. Il introduit la notion d'adaptateurs (figure 12) dans le but d'assurer une meilleure séparation entre les composants et d'offrir un gain pour la portabilité et la maintenance.

Prenons le cas de la couche la plus proche de l'utilisateur. Dans le modèle Arch, la Présentation correspond à la couche graphique physique de l'interface. La Présentation virtuelle définit les interactions au niveau logique. Par exemple, un choix multiple défini au niveau de la Présentation virtuelle pourra être physiquement proposé sous différentes formes à l'utilisateur (menu, boutons exclusifs, etc.) par le composant de présentation. Le fait de modifier la couche physique (soit en utilisant un *widget* différent, soit en changeant le type effectif d'interaction) n'affectera pas, idéalement, les autres composants du système.

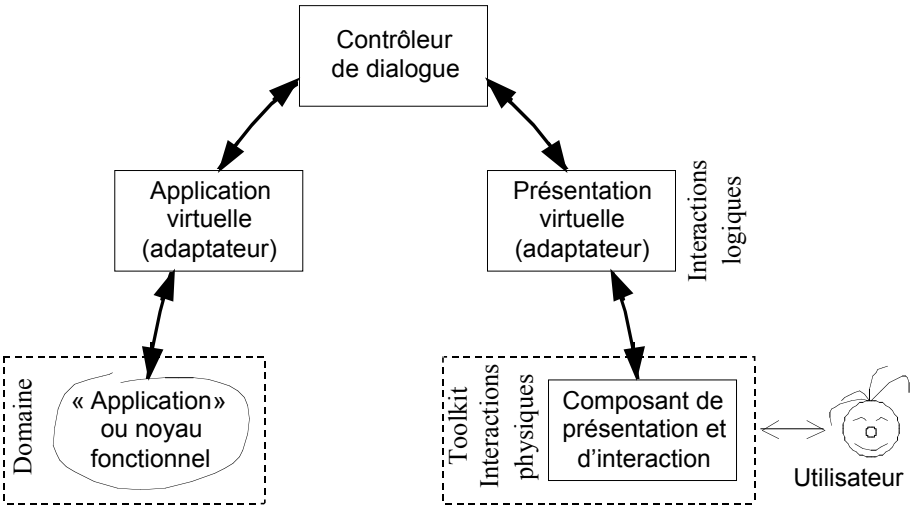


Figure 12 : le modèle Arch

MVC et PAC : les modèles multiagents[1]

Les modèles MVC et PAC structurent le système interactif en un ensemble d'agents qui travaillent en parallèle[2]. Un agent peut être par exemple une fenêtre, un *widget* voire, dans le cas des logiciels de dessin, un élément de la représentation graphique de l'objet d'intérêt, accessible à la manipulation directe. Il peut être aussi un service non directement accessible à l'utilisateur : un serveur FTP, une partie du programme en charge de la cohésion d'autres agents, etc.

Chaque agent possède une structure inspirée des modèles Arch ou Seeheim exposé précédemment. Il a une expertise, un état (i.e : une structure de données) et possède des émetteurs et récepteurs qui lui permettent d'acquérir et de générer des évènements à destination d'autres agents[3]. Ainsi, les modèles multiagents se caractérisent par une organisation fortement modulaire et hiérarchique et une communication par évènements [Coutaz.90].

Le modèle MVC (Model View Controller) est issu des recherches autour du langage à objet Smalltalk[4] à Xerox. Il s'organise autour de trois classes essentielles : la classe Modèle, la classe Vue et la classe Contrôleur [Olsen.97] [5], [Coutaz.90].

- La classe Modèle est la représentation abstraite de l'objet, c'est à dire le noyau fonctionnel au sens des paragraphes précédents.
- La classe Vue gère la présentation graphique du modèle. Plusieurs vues peuvent être associées au même Modèle.
- La classe Contrôleur est chargée de l'analyse des actions de l'utilisateur et de leurs répercutions sur les vues et sur le Modèle.

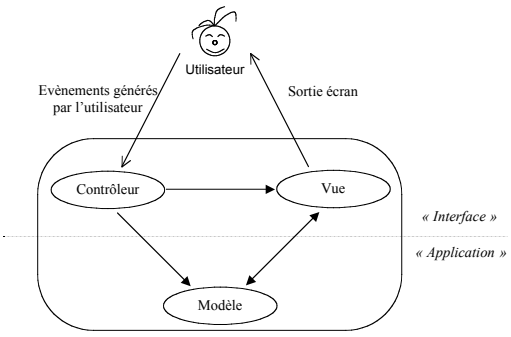


Figure 13 : le modèle MVC

[1] On trouvera dans [Coutaz.02] une description plus complète des modèles multiagents, ainsi que des principes heuristiques pour la conception de la hiérarchie des agents d'un système interactif.

[2] La notion de parallélisme ne doit pas être entendue ici au sens de « multiprocessus », mais comme une vue théorique – et ce même si les implémentations pratiques des architectures multiagents utilisent souvent les *threads* (processus léger). L'architecture multiagents est en outre indiquée lorsque le système doit utiliser plusieurs plates-formes, les agents communiquant alors à travers le réseau ([Coutaz.02]).

[3] Rappelons que le vocabulaire ici employé est théorique et n'induit pas à-priori une implémentation particulière. L'implémentation effective d'un agent reposera suivant les cas sur une classe, un module, un processus indépendant, etc, et celle de la communication inter-agents (émission et réception des évènements) sur des *pipes*, des messages, des appels de procédures ou des méthodes, etc.

Les langages de programmation objet sont particulièrement indiqués dans le cas des modèles multiagents, ne serait-ce que de par la protection des données qu'ils permettent. Les modèles multiagents peuvent cependant inspirer le développement d'un logiciel quand bien même le langage utilisé n'est pas orienté objet. C'est, incidemment, le cas de GENESIS qui est codé en langage C (voir chap. 11)

[4] C'est pourquoi la description du modèle MVC utilise la terminologie des langages orientés objets.

[5] Notons que dans ce même article, Olsen identifie quelques-uns des principaux problèmes inhérents à la conception des interfaces, tels que l'affichage, la mise à jour de l'affichage, la nécessité d'une optimisation, etc. Indépendamment de toute considération sur l'architecture logicielle, il offre des éléments de solution pour chacun d'eux.

La vue et le contrôleur constituent dans une architecture MVC l'interface utilisateur de l'agent. A un élément de l'interface (entendons : un objet visible, pas au sens du langage) correspond une instance de chaque type de classes. Les Vues s'enregistrent auprès du Modèle de sorte que lorsqu'un changement conséquent intervient dans ce dernier, il demande une mise à jour à chacune de ses vues. Ceci mis à part, le Modèle ne connaît rien de l'interface utilisateur : il fournit simplement un ensemble de services permettant de lire ou de modifier l'état de ses données. Idéalement, l'interface peut être modifiée sans que le noyau conceptuel, dans le modèle, n'en soit affecté. La vue et le contrôleur peuvent par contre avoir une connaissance explicite du modèle.

Un agent PAC [Coutaz.87] fait également apparaître trois composants, mais dont les rôles sont différents (figure 14) :

- L'Abstraction (A) est le noyau fonctionnel de l'agent, partie conceptuelle indépendante de toute représentation visuelle ou sonore. Plusieurs abstractions peuvent, le cas échéant, être regroupées à la manière d'une base de données.

- La Présentation (P) est en charge de toutes des interactions utilisateur en entrée (événement) comme en sortie. Plusieurs Présentations peuvent figurer dans le même agent pour proposer plusieurs visualisations de l'Abstraction à l'utilisateur.

- Le Contrôle (C) maintient la cohérence entre Abstraction et Présentation. Il répercute les changements intervenant sur A (modification des données) ou P (activation d'un bouton par exemple) et permet, le cas échéant, le changement de formalisme entre P et A. Il s'occupe également de la mise à jour des différentes Présentations de l'agent dans le cas d'une visualisation multiple. Enfin, il est en charge de la communication inter-agents : c'est par leur Contrôle que les agents sont connectés.

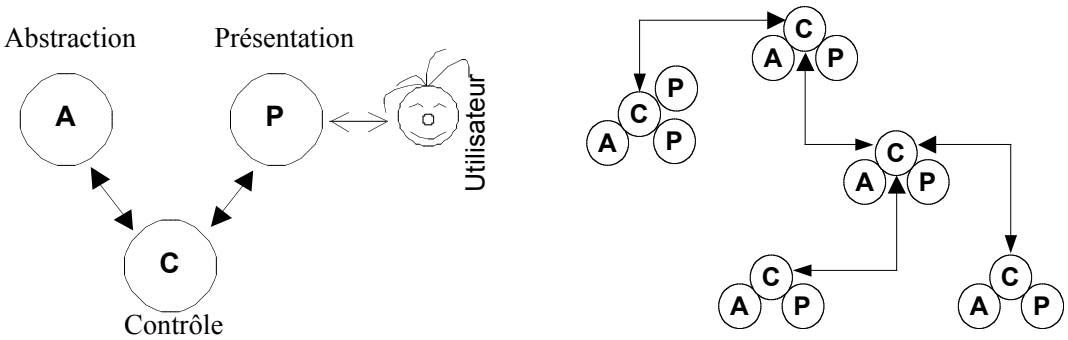


Figure 14 : agent PAC et hiérarchie d'agents PAC[1]

[1] Rappelons que les doubles flèches représentent les flux d'information, pas l'héritage : il n'y a pas d'héritage (au sens des langages orientés objets) entre agent, mais des liens de dépendance.

Les modèles d'architecture et le développement de GENESIS

La mise en œuvre pratique des modèles d'architecture est souvent difficile. Baecker (et d'autres auteurs) rappellent qu'il est parfois impossible de séparer en pratique l'interface du noyau fonctionnel, notamment dans le cas de la manipulation directe où interface et fonctionnalités sont intimement liées [Baecker&al.95d].

Par ailleurs, Coutaz montre que le rapport coût/bénéfice de la mise en œuvre d'une architecture logicielle conceptuelle est parfois défavorable [Coutaz.02]. C'est notamment le cas lorsqu'il s'agit de concevoir un prototype, qui par nature suppose un temps de développement réduit et comporte des parties entières appelées à être supprimées ou modifiées en profondeur. Or, si l'environnement GENESIS est diffusé auprès d'un nombre croissant d'utilisateurs, il constitue avant tout le résultat d'un travail de recherche. GENESIS se situe ainsi à mi-chemin entre *prototype* et *produit*.

De façon plus générale, les modèles d'architecture *n'ont pas* pour vocation d'être traduits in-extenso lors d'applications pratiques. Ils constituent des guides théoriques et doivent, en quelque sorte par nature, être déformés, adaptés, contournés lors de l'implémentation.

Suite à ces deux remarques, précisons que nous n'avons pas cherché à traduire en pratique l'un des modèles Seeheim, Arch MVC ou PAC précédemment évoqués de façon exclusive et précise. Chacun d'eux a constitué une source d'inspiration pour l'architecture de GENESIS pour ainsi dire au cas par cas – et ce quand bien même ils s'opposent sur certains points. Ce n'est pas là, incidemment, une démarche qui nous est propre. Selon Coutaz : « les concepteurs d'interface se doivent de considérer l'architecture sous des perspectives multiples. (…) Plusieurs structures se groupent pour constituer l'expression concrète d'une architecture »[1] [Coutaz.02].

Nous terminons ici ce parcours général autour des théories et concepts fondamentaux de l'Interaction Homme-Machine. La suite de cette partie est consacrée à l'exposé de principes, guides et méthodes plus pragmatiques, que nous aurons considérés au jour le jour pour la conception de GENESIS.

[1] " *Software designers must consider multiple perspectives on architectural design. (...) Multiple structures altogether constitute the concrete expression of an architecture* »

Chapitre 4

GUIDES ET PRINCIPES ERGONOMIQUES

Les 4 principes de Hansen (1971)[1],
Les principes de Norman(1983)[2],
Les 93 principes de Rubinstein (1984)[3],
Les « règles d'or » de Shneiderman[4],
Les 4 principes de Gould[5],
Les principes des industriels[6],
…

La littérature de l'Interaction Homme-Machine abonde en liste de principes et conseils plus ou moins empiriques. L'adoption d'une telle présentation n'est pas fortuite : elle est particulièrement efficace lorsqu'il s'agit d'indiquer de façon pragmatique au concepteur en bute à des problèmes pratiques les points à considérer et les éléments de solution à envisager.

Si les listes de principes présentent de nombreuses similitudes, elles s'opposent aussi parfois – ce qui n'est guère étonnant puisque des points de vue différents existent quant à ce qu'est une « bonne » interface. Les principes énoncés doivent toujours être adaptés aux situations concrètes ; Shneiderman appelle ainsi à ce qu'une large compilation soit effectuée pour chaque projet [Shneiderman.98 p89].

C'est notre propre compilation que nous exposons dans ce chapitre. La présentation adoptée est à l'image de celles que nous avons rencontrées dans les différents écrits : elle fait apparaître une liste, chaque paragraphe n'ayant pas nécessairement de rapport direct avec ses voisins.

4.1 - Les principes inspirés de Hansen.

Les quatre principes de Hansen sont, en 1971, les premiers à avoir été proposés. Leur contenu apparaît de façon récurrente dans la littérature. Nous les rappelons ici, en les élargissant le cas échéant de considérations plus récentes [Baecker&al.95c].

[1] Voir par exemple [Baecker&al.95c].
[2] [Norman.83]
[3] Voir par exemple [Baecker&al.95c].
[4] [Shneiderman.98]
[5] Voir notamment [Gould.88]
[6] Voir par exemple le site d'IBM, sur lequel figure un grand nombre de principes et guides [IBM].

4.1.1 - Connaître l'utilisateur et ce qu'il veut faire

La nécessité de connaître l'utilisateur n'est jamais mise en question dans la littérature[1]. Il convient d'avoir une idée précise :

- De l'expérience de l'utilisateur potentiel quant à l'usage de l'informatique ;
- De sa connaissance du domaine spécifique au système que l'on souhaite concevoir – c'est à dire, dans le cas spécifique à GENESIS, de sa connaissance de l'informatique musicale, de la Physique, et des instruments de musique traditionnels.

Précisons, donc, quelques caractéristiques de l'utilisateur auquel GENESIS s'adresse :

- Il connaît les principes habituels des interfaces WIMP ; il a une bonne pratique de l'outil informatique.
- Il n'a pas de connaissance particulière en Physique.
- Il ne connaît pas à priori le formalisme CORDIS-ANIMA.
- Son objectif est en règle générale la création sonore et musicale – sans présumer ni de l'importance ni de l'ambition de cette création.
- Son objectif peut être aussi, de façon secondaire, de découvrir ou de faire découvrir par la pratique certains concepts clés de la Physique élémentaire (inertie, raideur, mode de vibration...).

Remarque : le formalisme CORDIS-ANIMA et la *pensée physique* sont des idées nouvelles pour l'utilisateur débutant – comme, d'ailleurs, pour l'utilisateur expert en modèle de signal et non en modèle physique. Il est en conséquence nécessaire que GENESIS permette un apprentissage ou auto-apprentissage *par le faire* (disons : « enactif ») des possibilités du formalisme. Il est bien clair cependant que la fonction première dévolue à GENESIS reste la création. En conséquence, si l'apprentissage se doit d'être soutenu par l'organisation de GENESIS il devra l'être aussi par un *environnement pédagogique* complémentaire.

4.1.2 - Minimiser la mémorisation

Hansen appelle à ce que l'interface ne surcharge pas la mémoire de l'utilisateur. Ce principe était sans doute particulièrement important pour les interfaces en lignes de commande, qui tendent par nature à beaucoup solliciter la mémoire, mais reste tout à fait valable dans le cas des interfaces graphiques. La suite de ce chapitre présentera plusieurs conseils que nous avons suivis afin de respecter cet important principe (voir notamment l'analyse de Norman et la notion de *visibilité*).

4.1.3 - Optimiser les opérations

Hansen conseille également de faciliter et d'optimiser les opérations courantes, tant en termes ergonomiques qu'en termes de vitesse d'exécution. Dans ce but, nous avons systématiquement cherché à hiérarchiser les fonctionnalités en fonction de la fréquence probable de leur utilisation et à adapter en conséquence leur ergonomie respective. Nous avons par ailleurs eu recours à plusieurs reprises à une démarche inspirée du modèle GOMS (paragraphe 3.2.2 -).

---

[1] Les premiers des 30 « éléments de conception » (design elements) de Heckel y font par exemple référence. Heckel appelle entre autre à ce que l'interface « parle le langage de l'utilisateur » (« *6 - Speak the user langage* ») - in [Baecker&al.95c].

4.1.4 - Gérer correctement les erreurs

La prise en compte du risque d'erreur est le dernier point soulevé par Hansen[1]. Cela implique, d'abord, que les choix ergonomiques effectués tendent à les minimiser, ensuite qu'une gestion correcte des erreurs inévitables (de l'utilisateur mais aussi, le cas échéant, du programme) soit mise en œuvre. Entre autres choses, il convient de concevoir des messages explicites, de rendre les actions réversibles, de permettre une récupération de l'état du système en cas de fermeture inopinée du programme, etc.

4.2 - Modèle mental, *affordance*, contrainte, *mapping*, visibilité et retour : l'analyse de Norman

En parallèle à sa Théorie de l'action, Norman a introduit d'autres concepts relatifs au rapport de l'homme à ses outils, à la façon dont il les aborde et interagit avec eux. Dans l'ouvrage [Norman.88], il s'intéresse d'abord à de nombreux objets de la vie courante (Ciseaux, porte, réfrigérateurs, téléphones, etc.) pour montrer ce qui fait la qualité et les défauts de leur ergonomie. L'analyse est cependant parfaitement appliquable dans le cadre de l'interaction homme-ordinateur. Elle nous a offert un cadre de réflexion pertinent. Nous la résumons dans ce paragraphe, qui introduit successivement les concepts de modèle mental, *d'affordance*, de contraintes, de *mapping* (projection), et, enfin, de visibilité et retour (*feedback*).

4.2.1 - Modèle mental

Le fonctionnement des outils s'accompagne toujours de conceptualisations de diverses natures chez leurs utilisateurs. Le modèle mental (ou modèle conceptuel) de l'utilisateur, c'est précisément la représentation mentale qu'il se fait du fonctionnement d'un système (figure 15 page suivante). Comme l'explique Norman, lorsque qu'une personne utilise un système (ou un outil), elle ne s'appuie pas sur ses propriétés *réelles* (son *modèle interne*) mais sur le modèle mental qu'elle s'en fait. C'est le modèle mental qui lui permet, par exemple, de prédire l'effet de ses actions ou encore d'interpréter leurs résultats.

Le modèle mental dépend de multiples facteurs, tel le bagage initial de l'utilisateur quant au domaine de l'outil. Mais il est important de comprendre que c'est essentiellement au contact de l'outil lui-même qu'il se construit – et non, incidemment, par la lecture des différents manuels qui l'accompagnent éventuellement.

Il importe avant tout que le modèle mental soit *opératoire*, c'est à dire notamment qu'il soit suffisamment clair et simple et qu'il permette à l'utilisateur *d'anticiper les conséquences de ses actions* : ce que croit faire l'utilisateur doit correspondre à ce qu'il fait.

Norman note que le modèle mental est parfois « superstitieux » (i.e. : très différent du modèle interne), et que cela pose assez souvent problème[2]. Aussi, Beaudouin-Lafon

[1] C'est aussi l'une des six « règle d'or » de Shneiderman, qui propose quelques clés pour l'observer en pratique [Shneiderman.98 pp75-79]

[2] Ce n'est cependant pas toujours le cas – qu'on songe, par exemple, à la représentation mentale communément associé au fonctionnement d'une voiture. L'essentiel reste, en définitive, que le modèle mental soit opératoire. D'aucun, d'ailleurs considère qu'une partie du travail du concepteur consiste précisément à permettrent aux utilisateurs d'utiliser le système sans avoir a comprendre son fonctionnement – donc à ce qu'un modèle mental superstitieux se développe. Nous verrons que ce n'est pas l'option que nous avons adoptée (voir également la notion de métaphore dans le paragraphe suivant).

appelle à ce qu'une certaine identité existe entre le modèle interne du système et le modèle mental qu'il induit. Pour lui, une interface que l'on comprend *« c'est une interface dont le modèle interne correspond au modèle mental de l'utilisateur »* [Beaudouin.99].

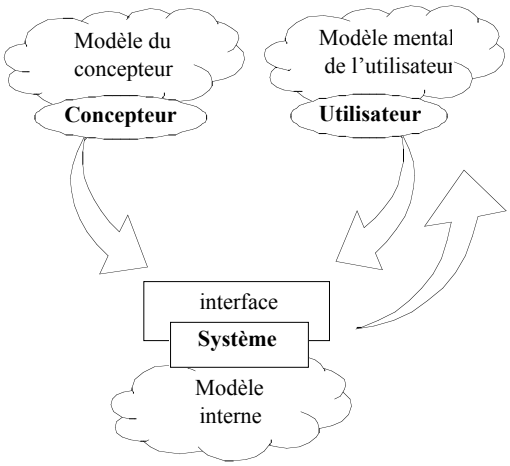


Figure 15 : modèles conceptuels.
D'après Norman, [Norman.88].

En plus du modèle interne du système et du modèle mental de l'utilisateur, Norman évoque le modèle du concepteur (figure ci contre). Le modèle du concepteur est une représentation relativement claire et fidèle du fonctionnement du système. C'est donc, précisément, un modèle *opératoire.*

Aussi, selon Neale : « dans un cas idéal, le modèle de l'utilisateur sera compatible avec le modèle du concepteur »[1] [Neale&Carroll.97]. Un objectif possible pour le concepteur – que nous aurons parfois considéré pour GENESIS – est donc de chercher à communiquer par l'interface, son *design* et son ergonomie, son propre modèle à l'utilisateur.

La notion de modèle mental aura occupé une place centrale tout au long de nos travaux et il y sera fait référence à plusieurs reprises dans la suite du document. Incidemment, l'intérêt majeur d'un grand nombre des principes énoncés plus bas tient à ce qu'ils peuvent aider le concepteur d'interface à communiquer ou induire *via* l'interface un modèle mental spécifique.

4.2.2 - *Affordance*

Le concept *d'affordance* (littéralement : « offrande », mais la traduction n'est pas satisfaisante et le terme anglais est conservé dans la suite) a d'abord été introduit par Gibson [Gibson.79]. L'environnement présente aux animaux qui y vivent des dangers ou au contraire des bénéfices. Ces *affordances*, selon Gibson, constituent une propriété intrinsèque des objets. Par exemple, un prédateur *est* la mort, une nourriture la vie, etc. Il importe que l'animal soit armé pour percevoir les *affordances* de l'environnement. Une hypothèse essentielle de Gibson est que les *affordances* sont présentes dans le « réseau optique ambiant », donc perceptibles par le canal visuel.

Norman reprend dans son travail la notion d'*affordance* de façon plus pragmatique. Pour lui, *l'affordance* d'un objet correspond à ses qualités perceptibles qui indiquent à la fois sa fonction et son « mode l'emploi ». Par exemple, les ciseaux *afford* (offrent) de saisir et de couper, la chaise incite à s'asseoir, etc.

Norman illustre le concept avec l'exemple du téléphone. Les téléphones modernes ont de multiples fonctions à priori très utiles : mise en attente, secret, rappel automatique, répertoire, etc. Certains d'entre eux, cependant, ne les mettent pas en évidence de façon visible. Ils deviennent alors inutilisables, et ce quand bien même leur utilité *potentielle* est grande. Un problème d'un autre type peut survenir, note Norman, lorsque les *affordances* ne correspondent pas aux propriétés réelles de l'objet.

[1] Traduction de l'auteur - « Ideally, the user's model will develop to be compatible with the designer's model ».

Dans les interfaces WIMP, les *affordances* de plusieurs *widgets* sont assez claires : un bouton appelle le clic, un menu la sélection, etc – on pourrait parler, à ce sujet, « *d'affordances* élémentaires ». On peut cependant tout aussi bien discuter de *l'affordance* d'une fenêtre, de la représentation de l'objet d'intérêt ou encore de l'interface dans son ensemble, etc.

Ainsi, le concept *d'affordance* est très utile pour la conception des systèmes interactifs, qu'il s'agisse par exemple de choisir les *widgets*, leur organisation, ou encore la représentation de l'objet d'intérêt. Il est aussi essentiel pour l'évaluation. Il permet de comprendre, par exemple, pourquoi certaines interfaces sont utilisables sans formations préalables alors que d'autres posent un problème constant aux utilisateurs, toujours dans l'expectative quant la fonction et le mode d'emploi de telle ou tel partie.

Le notion *d'affordance*, et le conseil suivant lequel il convient de promouvoir les *affordances* dans les interfaces, ont occupé une part importante dans notre réflexion. Ils ont guidé, par exemple, les choix opérés quant à la représentation des objets CORDIS-ANIMA. Nous avons plus généralement cherché, à tous les niveaux de l'interface, à dégager la solution qui *afford* le plus – entendons : qui rend perceptible à la fois la fonction *et* le « mode d'emploi ».

4.2.3 - Contraintes

Certains objets contraignent l'utilisateur en interdisant certaines actions. Par exemple, on peut considérer qu'une paire de ciseau contraint l'utilisateur à ne pas se saisir des lames – et qu'à l'inverse elle invite à ce que les poignées soient saisies. Les contraintes d'un outil, selon Norman, sont les facteurs physiques, culturels ou logiques qui découragent les utilisations erronées, et de là encouragent une utilisation correcte. Elles sont essentielles en ce qu'elles guident la progression de l'utilisateur face à l'outil.

4.2.4 - Visibilité et retour (*feedback*)

Norman attire également l'attention sur l'importance de la visibilité (fait de rendre visibles les objets et actions possibles) et du retour des actions entreprises. La visibilité est essentielle pour une bonne communication du modèle interne du système. Le retour (notamment visuel) que produit le système en réponse aux actions de l'utilisateur est quant à lui nécessaire pour qu'il puisse être certain de leur réalisation effective et conscient de leurs effets[1].

Les principes de *visibilité* et *retour* se concrétisent dans GENESIS par exemple par les changements de couleurs de certains boutons en fonction de l'état de l'interface ; par la visualisation dynamique de la représentation des objets CORDIS-ANIMA ; par le « flash » visuel qui marque tout enregistrement sur disque. De façon plus générale, nous résumerons leur mise en application par la formule suivante : dans GENESIS, nous avons cherché à *donner à voir ce qu'on peut faire et ce qu'on fait* et à *permettre de faire ce qu'on voit*.

[1] Incidemment, une difficulté inhérente aux modeleurs / simulateurs tels que GENESIS est que le retour *terminal* des actions (i.e. : leurs conséquences sur les phénomènes qui seront finalement obtenus) ne *peut pas* être rendu perceptible immédiatement : il nécessite dans tous les cas une simulation. Nous reviendrons sur ce point au paragraphe 4.7.2 - p101.

4.2.5 - Projection (*Mapping*)

Les *mappings* sont les relations qui existent entre un contrôle et ses effets sur le système. Un exemple particulièrement réussi de *mapping* présenté par Norman est celui de la direction des automobiles : si l'apprentissage du code de la route est difficile, la manipulation du volant est, elle, particulièrement intuitive.

4.2.6 - Les conseils de Norman

Nous considérons avec Norman qu'*affordance*, contrainte, *mapping* visibilité, retour et modèles mentaux figurent parmi les concepts fondamentaux régissant l'interaction de l'homme avec ses outils.

A la fin de l'ouvrage [Norman.88], Norman y fait référence pour proposer sa *méthode de conception centrée sur l'utilisateur* (*user-centered design*) et plusieurs conseils utiles au concepteur d'interface. Ces conseils sont reproduits in-extenso ci-contre.

- *Faciliter la détermination [par l'utilisateur] des actions possibles à tout moment – avoir recours aux contraintes.*
- *Utiliser le canal visuel. Rendre apparent le modèle conceptuel du système, les actions alternatives et les résultats des actions.*
- *Rendre aisée l'évaluation de l'état du système quel qu'il soit.*
- *Utiliser des mappings naturels.*

En d'autres termes, s'assurer que :
1) l'utilisateur puisse déterminer ce qu'il est possible de faire.
2) l'utilisateur puisse expliquer ce qui est en train de se passer.

Figure 16 : quelques conseils d'après [Norman.88] (traduction libre)

4.3 - Métaphores

De façon générale, une métaphore est une analogie ou encore une association d'idées entre un *domaine source* et un *domaine cible*.

Notre cerveau fonctionne pour une grande part avec de telles analogies et associations et il semble que les métaphores soient essentielles à l'ensemble de nos activités cérébrales. Par exemple, dans le langage courant, l'utilisation des métaphores est la règle et non pas l'exception [Erickson.95]. Ainsi, on *met une idée en lumière* ou *en perspective* comme on le ferait d'un objet – mais les exemples pourraient être multipliés. Nous évoluons, en fait, dans un univers de métaphores implicites sans même en avoir conscience.

La notion de métaphore a profondément marqué l'histoire de l'Interaction Homme-Machine.

Dans une interface une métaphore représente le système en général à l'aide d'un objet de la vie quotidienne. Elle se place à la « surface » du système (à son *interface*) et structure l'ensemble de l'interaction. Le « bureau » (*desktop*) de nos ordinateurs, qui fait

apparaître des « dossiers », une « corbeille » et permet de « couper » ou « coller », est une vaste métaphore pour les systèmes de fichiers – sans doute la plus célèbre en ergonomie. Son apparition au début des années 80 (d'abord à Xerox puis dans le système macintosh) aura révolutionné les interfaces [Neale&Carroll.97]. Le courrier électronique est un autre exemple de métaphore plus récent.

Pour Bruner, les métaphores constituent une passerelle essentielle pour l'apprentissage, permettant qu'un savoir préalablement acquis sur le domaine source puisse être appliqué sur le domaine cible. Selon Neale, l'intérêt des métaphores dans le contexte de l'IHM se situe pourtant bien au-delà de l'apprentissage : elles facilitent l'ensemble de l'utilisation du système en structurant le modèle mental de l'utilisateur [Neale&Carroll.97].

D'autres considèrent que les métaphores présentent un danger en ce qu'elles remplacent la *compréhension* par *l'approximation* [Nelson.90]. C'est que les associations entre domaine source et cible créées par nos métaphores sont le plus souvent cachées et peu précises. Pour Neale, ces critiques ne tiennent plus dès lors que l'utilisateur a conscience de ce que la métaphore n'est qu'un moyen commode d'envisager le système, un raccourci. Il ajoute que le flou de la métaphore constitue l'une des forces de la métaphore : elle évite à l'utilisateur l'apprentissage d'un modèle abstrait qui s'ajouterait à la tâche qu'il veut réaliser[1]. Ainsi, la métaphore avantagerait les utilisateurs dont le mode de pensée et d'apprentissage prédominant est *concret* ou *visuel*, au détriment des *conceptuels* et elle se place dans la droite ligne de l'ensemble des interfaces WIMP. Toujours selon Neale, la puissance du concept est telle que l'utilisateur développe *dans tous les cas* une approche métaphorique des systèmes auxquels il est pour la première fois confronté.

A la fin des années 80 la notion de métaphore occupait une place centrale parmi les outils de l'ergonome[2]. Le mot d'ordre, alors, était de rechercher La métaphore qui permettrait de cacher le fonctionnement de la machine (ou du logiciel) pour que l'utilisateur puisse avoir l'impression d'évoluer dans un contexte plus familier – ou du moins la représentation d'un tel contexte. Dès lors qu'elle était choisie, il était conseillé pour qu'elle offre tous ses avantages de chercher systématiquement à l'étendre à tous les aspects de l'interface [Erickson.90]. Et cela, parfois, jusqu'au blocage… Selon Nelson, justement, les métaphores ont trop souvent tendance à constituer un « poids mort » pour le concepteur, qui se croit obligé de leur conditionner l'organisation de l'ensemble de l'interface. Aussi, Nelson se réjouit de ce que « l'idéologie métaphorique » ait aujourd'hui repris une plus juste mesure [Nelson.90].

Comment se situer, pour la conception de GENESIS, face à la vaste discussion qui a animé l'Interaction Homme-Machine autour de la notion de métaphore ? Faut-il déduire des avantages incontestables du bureau et d'autres métaphores réussies qu'il est préférable pour faire une « bonne » interface d'implémenter une métaphore ?

Le statut des métaphores dans GENESIS se présente en trois temps.

Tout d'abord, il faut rappeler que le formalisme CORDIS-ANIMA autorise un modèle mental à la fois fidèle à son *modèle interne* et opératoire. Un rôle qui nous semble

[1] Pour Neale, une métaphore se caractérise toujours *à la fois* par sa fidélité (match) et son infidélité au domaine source (mismatch) [Neale&Carroll.97]. Selon Kay, l'infidélité de la métaphore est un avantage : une métaphore se doit d'être « magique » en permettant de réaliser dans le domaine cible des opérations qui ne sont pas possibles dans le domaine source [Kay.90].

[2] Erickson rapporte ainsi que la notion de métaphore passait dans les années 80 pour être le « graal sacré » d'Apple [Erickson.95].

essentiel pour GENESIS, alors, est que l'interface induse ce modèle mental ainsi qu'une bonne compréhension du formalisme. GENESIS ne doit donc pas cacher le *modèle interne* de CORDIS-ANIMA, mais au contraire le révéler. Pour cette raison, nous avons décidé de laisser apparent les modules définis par le formalisme, de permettre une manipulation directe des objets, et surtout de *ne pas* chercher à organiser GENESIS autour d'une métaphore.

Ceci dit, nous avons tout de même voulu qu'une certaine analogie puisse s'établir entre GENESIS et un atelier de lutherie d'une part, entre l'espace de construction des objets et l'établi du luthier. Les termes utilisés dans l'interface – « objet », « établi », « lutherie », etc. – auront autant que possible été choisis en ce sens.

En nous basant sur cette similitude de surface, nous dirons dans la suite du document que GENESIS propose une métaphore de l'atelier du luthier. Mais attention : ce n'est pas au sens ou on l'entend dans la littérature. Notamment, la métaphore de l'atelier ne vise en rien à « cacher le modèle interne » en substituant au domaine cible (CORDIS-ANIMA) le domaine source (la lutherie traditionnelle). Ce serait là, d'ailleurs, un objectif absurde puisque le travail du luthier n'est pas en général connu de l'utilisateur de GENESIS, et qu'il ne pourrait s'en inspirer pour guider en profondeur son travail. L'intérêt de la métaphore de l'atelier de lutherie dans GENESIS est donc moins ambitieux, bien qu'essentiel. Elle vise en fait à ramener l'utilisateur au réel et au matériel et surtout à soutenir l'idée que l'on *construit* ces objets (avant de les mettre en jeu).

En définitive, il s'avère difficile de situer de manière précise et répertoriée la notion de métaphore au sein de GENESIS: Implicite ou explicitée, elle est de toute façon présente à peu près à tous les niveaux. Pour n'en citer qu'un : le modèle physique lui-même n'est-il pas une métaphore de la réalité de l'instrument ? Ainsi, les principales métaphores ne sont pas tant dans l'interface que dans le formalisme qui en est le fondement (chaque module étant métaphorique de son vis-à-vis réel) et plus encore dans le travail de l'utilisateur lorsqu'il veut, avec le *langage* CORDIS-ANIMA, exprimer des métaphores d'objets réels.

4.4 - Economie, simplicité et souci du détail

Il importe, lorsqu'on conçoit une interface, de chercher à limiter la charge cognitive induite par son utilisation. La dernière des huit « règles d'or » de Shneiderman demande, par exemple, que la mémoire à court terme de l'utilisateur ne soit pas surchargée. D'autres auteurs appellent à ce que le débit d'information visuelle soit limité – mais le principe d'économie est en fait très général.

Nous savons, d'expérience, combien le concepteur d'interface tend naturellement à la surcharger– voulant ainsi, peut être, impressionner par sa complexité. Ce n'est pourtant pas une bonne stratégie pour « économiser l'utilisateur ». Il convient au contraire de rechercher systématiquement la simplicité au sein de l'interface, par exemple au niveau de l'organisation des fenêtres ou des raccourcis, de la présentation des concepts, si ce n'est au niveau des concepts eux-mêmes.

Une attitude efficace consiste à délibérément chercher à *faire le plus* avec le *moins*. Norman appelle ainsi à ce qu'on se demande plusieurs fois combien de boutons et contrôles sont *effectivement* nécessaires avant de les implémenter : moins il y a de contrôles à comprendre et apprendre, plus l'interface paraît (et est) facile d'utilisation [Norman.88]. On notera cependant que ce conseil peut être en désaccord avec l'injonction

du même auteur appelant à *rendre les choses visibles* ; un compromis doit, en la matière, être trouvé.

Le principe d'économie s'étend à d'autres aspects de l'interface. Il concerne par exemple le nombre de fenêtres, le nombre de couleurs, le nombre de polices de caractères (voir plus bas), voire, dans certains cas, la diversité des fonctionnalités disponibles. Attention cependant : le principe d'économie ne *s'applique pas* à l'utilisation des ressources de la machine ou à la complexité interne du programme. Si une certaine optimisation du code doit être recherchée puisqu'elle est nécessaire à la fluidité de l'interface, des architectures complexes peuvent contribuer à l'utilisabilité du système.

Enfin, nous avons accordé une attention particulière aux *détails* de GENESIS. Un détail mal conçu, en effet, fait rapidement apparaître l'interface comme un obstacle, alors qu'on souhaite d'une certaine manière qu'elle disparaisse de la perception de l'utilisateur au profit de la tâche à accomplir. Ainsi, tout défaut élémentaire devient rapidement insupportable à celui qui utilise couramment l'interface[1].

4.5 - Cohérence et conformité

Cohérence et conformité sont des principes très souvent mis en exergue dans la littérature.

Le principe de cohérence (*internal consistency*) apparaît ainsi en tête des huit « règles d'or » de Shneiderman [Shneiderman.98]. Il consiste en ce que les mêmes conventions et règles soient conservées à travers l'ensemble de l'interface.

La cohérence se décline sous plusieurs formes. On peut distinguer notamment : la cohérence *d'action* (il s'agit qu'à des situations similaires correspondent des séquences d'actions similaires) ; la cohérence de *langue* (la même terminologie doit être employée dans l'ensemble de l'interface) ; la cohérence *d'organisation* (les différentes fenêtres doivent respecter une charte d'organisation) ; la cohérence graphique et typologique (police de caractères, aspect des *widgets*, couleurs, etc.). La charte graphique de GENESIS est un exemple simple de notre volonté de parvenir à une bonne cohérence.

Le principe de *conformité* (*external consistency*) demande quant à lui que le système soit mis en conformité avec les conventions culturelles des interfaces graphiques et plus encore avec les environnements informatiques auxquels l'utilisateur est habitué compte tenu de son domaine d'expertise. Marcus indique, par exemple, qu'il est souvent préférable de baser les choix ergonomiques sur l'expérience des utilisateurs plutôt que de mettre en œuvre une technique nouvelle qu'il lui faut apprendre, et ce même si celle-ci offre une meilleure efficacité [Marcus.97].

Les concepteurs de certains systèmes courants ont diffusé des règles pour encourager les développeurs à observer une certaine conformité[2]. Cependant, les logiciels tournant sur le système IRIX de Silicon Graphics sur lequel GENESIS a été développé ne sont guère standardisés. Qui plus est, IRIX n'est pas le système auquel les utilisateurs potentiels de GENESIS sont majoritairement habitués. Aussi, pour appliquer le principe de conformité,

[1] Ce peut être, par le temps de réaction des menus, un problème d'actualisation des données, une petite difficulté récurrente pour la saisie de certains objets graphiques par manipulation directe de l'objet d'intérêt, une mauvaise gestion de la superposition des fenêtres ou tout autre problème qui n'aura pas été envisagé lors de la conception globale de l'interface.

[2] Citons, par exemple, les règles d'Apple : « Apple Computer : Macintosh Human Interface Guidelines », Addison-Wesley 1992 ». Ces règles ont sans aucun doute largement contribué à l'homogénéité relative des différents logiciels tournant sur la plateforme, et par suite à leur utilisabilité et leur succès.

nous aurons essayé de nous inspirer des standards des systèmes Macintosh et Windows – étant entendu que l'apparence et le comportement (le « *look and feel* ») du gestionnaire de fenêtres proposé par IRIX en est fort éloigné. L'organisation des principaux *items* des menus Fichier et Edition est un exemple de *mise en conformité* au sein de GENESIS[1].

4.6 - Quelques conseils pour une bonne communication visuelle

Dans les interfaces graphiques, le canal visuel est le principal canal sensoriel par lequel transite l'information du système vers l'utilisateur[2]. Il est essentiel de comprendre son fonctionnement (c'est la *théorie* de la perception visuelle dont nous avons exposé quelques aspects au paragraphe 3.2.1) et plus encore de disposer de quelques guides et principes avisés qui permettent de tirer parti de ses propriétés. C'est l'objet de ce paragraphe.

4.6.1 - Les conseils de Marcus

Marcus propose plusieurs conseils, de bon sens il est vrai, mais qui nous aurons aidé dans notre recherche d'une bonne communication visuelle [Marcus.95], [Marcus.97]. Ils s'organisent autour de trois injonctions : organiser, économiser, et communiquer.

Organiser

Organiser l'écran permet de soutenir la capacité humaine à reconnaître les structures. Une bonne organisation suppose :

- Que le principe de cohérence soit observé - des informations similaires devant être visualisées de manière similaire.
- Que des grilles soient utilisées pour aligner les différents objets dans les fenêtres.
- Que les diverses *relations* existant entre différents objets soit soulignées visuellement. La proximité, l'encadrement, la couleur peuvent y aider.

Enfin, pour parvenir à une bonne organisation de l'affichage, il faut se rappeler que l'œil opère face à une image un parcours dynamique ; il convient donc de chercher à guider la navigation de l'œil en attirant l'attention sur les informations importantes et en limitant l'impact visuel des informations secondaires.

Economiser

Economiser en matière de communication visuelle, c'est selon Marcus :

- Limiter la diversité des objets graphiques utilisés ; l'économie se doit donc d'être considérée lors de la conception de la charte graphique.
- Rechercher une certaine modestie dans l'apparence (le « *look* ») des différents objets graphiques.
- Limiter le nombre et la complexité des informations portées à l'écran – sans pour autant nuire à la lisibilité ou induire une surcharge de la mémoire.

---

[1] Marcus ajoute à la cohérence et la conformité (internal and external consistency) le principe de « *real world consistency* » : l'interface doit être cohérente avec notre expérience de tous les jours [Marcus.97). Nous renvoyons à ce sujet au paragraphe consacré aux métaphores.

[2] Le recours au son, voire au canal gestuel (par l'emploi de périphériques à retour de force) constituent encore l'exception. GENESIS ne les utilise pas.

Communiquer !

Marcus rappelle d'abord que pour *communiquer* visuellement, il convient de ne pas oublier les déficiences du système visuel, qui sont assez courantes – et de considérer par exemple que la capacité à distinguer les détails peut décroître considérablement avec l'âge. Le choix des polices de caractères doit permettre une bonne *lisibilité* ; chacune d'entre elles devrait être employée dans un contexte bien délimité qui lui est propre.

Par ailleurs, pour transmettre plus vite et mieux des informations diverses il convient de reconnaître la puissance du *symbolisme* que peuvent véhiculer les images. La conception graphique est un sujet vaste et complexe [Baecker&al.95c]. Nous aurons quant à nous par exemple apporté un soin particulier à la conception des quelques icônes figurant dans GENESIS.

Enfin, Marcus observe que la communication des informations peut gagner en efficacité quand *plusieurs représentations* d'un même objet ou concept cohabitent et en font ressortir différents niveaux de détails ou d'abstractions. Nous discuterons dans la partie V de l'éventualité d'une trans-représentation des objets CORDIS-ANIMA dans GENESIS.

4.6.2 - De l'utilisation de la couleur[1] dans les interfaces graphiques

La couleur est très efficace pour soutenir la communication visuelle. Elle est particulièrement utile pour [Marcus.95], [Marcus.97], [Murch.95], [Salomon.90] :

- Attirer le regard ; demander une réponse ; faire ressortir une situation d'urgence[2] ;
- Souligner des relations entre plusieurs objets (similitude ou différence).
- Faire ressortir des parties dans un document ; permettre à l'utilisateur de marquer un document ([Salomon.90]) ;
- Ajouter des dimensions pour le codage visuel d'informations – la couleur est notamment efficace pour mettre en évidence des catégories.
- Augmenter l'agrément de l'interface.

L'utilisation de la couleur est cependant délicate, ne serait-ce que dans la mesure où sa perception est complexe[3].

Les trois injonctions de Marcus précédemment exposées (organiser, économiser, communiquer) s'appliquent à la couleur. Le principe d'économie s'exprime alors de manière très concrète : il est souhaitable, selon Marcus, de limiter le nombre de couleurs dans une interface à 5 +/- 2. Cela correspond à la capacité de mémorisation à court terme[4] et permet d'éviter une surcharge inutile du canal visuel.

Un autre conseil en matière d'utilisation de la couleur est de ne l'envisager que comme une touche finale renforçant une information visuelle déjà présente. Ce conseil vaut pour la prise en compte des utilisateurs daltoniens, mais a cependant des raisons d'être plus profondes. Une démarche pratique consiste à concevoir d'abord l'interface en niveaux de gris. Nous aurons le plus souvent observé ce conseil. La représentation graphique des

---

[1] Par *couleur* on entend ici une combinaison {teinte, intensité, saturation}.

[2] A ce titre, notons que la teinte *rouge* (couleur du sang) semble être la seule dont la symbolique soit universellement partagée par les différentes cultures.

[3] Voir le paragraphe 3.2.1 - p 69. [Murch.95] offre de nombreux conseils utiles directement déduits des données psychophysiques concernant la perception de la couleur. Il explique par exemple, que les teintes de courte longueur d'onde (bleu, etc.) doivent être évitées pour la représentation de détails fins, que les *verts* et les *rouges* sont mal perçus lorsqu'ils figurent sur les bords d'un écran relativement large – on préfèrera, alors, utiliser des bleus ou des jaunes, etc.

[4] Notons que le nombre 5+/-2 est encore inférieur au « nombre magique » de Miller.

objets CORDIS-ANIMA y échappe cependant. Considérant que la couleur est particulièrement efficace pour communiquer des catégories, nous avons décidé de l'utiliser pour coder les différents types de modules CORDIS-ANIMA[1].

Enfin, Murch observe qu'il y a trop de variables impliquées dans la constitution et la perception des couleurs pour que des règles effectives et précises soient édictées. Il conclut « de toutes façon, expérimentez » [Murch.95].

Cette injonction, ici formulée relativement à l'emploi de la couleur, mérite d'être très largement généralisée. Elle aura largement inspiré le déroulement de nos travaux et nous la retrouverons un peu plus loin dans le chapitre suivant consacrée à la méthodologie.

4.7 - Objet d'intérêt et manipulation directe : « Moins d'interface pour plus d'interaction »[2]

4.7.1 - Lorsque sont trahis les principes de la manipulation directe…

Beaudouin-Lafon reconnaît les vertus de la manipulation directe pour une meilleure interactivité des interfaces. Il observe cependant que ses principes fondamentaux se sont dissouts au fur et à mesure que les tâches qu'on voulait réaliser se sont complexifiées [Beaudouin.99].

Dans la majorité des interfaces, et notamment dans les plus usuelles, l'objet d'intérêt est très souvent caché et noyé par une multitude *d'objets d'interfaces* : fenêtre, icône, menus, palette, etc [Beaudouin.97]. Pour réaliser la plus grande partie des opérations intéressantes permises par le logiciel, il devient nécessaire d'avoir recours à une manipulation indirecte de l'objet d'intérêt à l'aide d'une nouvelle fenêtre dans laquelle un style d'interaction très différent est proposé. Cette fenêtre, de plus, est souvent *modale :* elle interdit toute interaction avec le document tant qu'elle n'est pas fermée.

Dès lors, les avantages essentiels de la manipulation directe sont perdus. L'interaction devient tout aussi « indirecte » que les interfaces en ligne de commande. L'attention de l'utilisateur est détournée de l'objet d'intérêt et de la tâche essentielle qu'il souhaite réaliser.

Considérant ces problèmes, Beaudouin-Lafon appelle à un retour aux principes originels de la manipulation directe. Il demande que les interfaces utilisateur fassent apparaître *moins d'interface* pour permettre *plus d'interaction* avec l'objet d'intérêt [Beaudouin.99].

Plusieurs solutions technologiques peuvent offrir un accès plus direct aux fonctionnalités et donc aider à relever un tel défi : les *tool-glass*, les *marking menus*, la reconnaissance de gestes en sont des exemples. Nous expliquerons plus loin pourquoi nous n'avons que très partiellement fait appel à de telles techniques, pour rester dans des styles d'interaction plus traditionnels.

Les remarques de Beaudouin-Lafon restent cependant valides dans ce cadre. Pour les mettre en œuvre en pratique, nous avons cherché :

[1] Voir le chapitre 12, partie V. Notons ici que les couleurs des modules sont entièrement reconfigurables dans GENESIS, ce qui permet de contourner en grande partie les problèmes que ce choix peut poser dans les cas de daltonisme.

[2] Ces mots sont extraits du titre de l'article [Beaudouin.99]

- A ce que l'objet d'intérêt soit, d'abord, clairement identifié et reconnu comme tel. Ce qui intéresse l'utilisateur ce *n'est pas* l'interface, mais l'objet qu'elle lui permet de manipuler et la tâche qu'il souhaite réaliser.
- A ce que l'attention de l'utilisateur soit autant que possible focalisée sur cet objet.
- A maximiser le nombre des opérations qui se font en interaction directe avec l'objet d'intérêt sans passer par des objets interfaces ;
- A minimiser le nombre et l'importance (graphique notamment) des objets d'interface[1].
- A éviter, en outre, que les fenêtres additionnelles malgré tout nécessaires ne soient pas bloquantes, de telle sorte qu'elles puissent être envisagées par l'utilisateur comme un complément d'information ou de possibilités, et non comme un changement momentané et radical du style d'interaction.
- A considérer qu'à toute fenêtre additionnelle correspond un nouvel « objet d'intérêt local », et en conséquence à mettre en œuvre les propositions précédentes de façon récursive.

4.7.2 - La manipulation directe dans les modeleurs/simulateurs

Nous évoquons ici quelques-uns des problèmes inhérents à la manipulation directe dans le cas des interfaces qui, comme GENESIS, font appel à un processus de modélisation / simulation.

Une première remarque essentielle tient à la distinction qu'impose ce type de processus entre *modèle* et *résultat*. On a vu plus haut comment la manipulation directe a conduit au slogan WYSIWYG (What You See Is What You Get, ce que vous voyez et ce que vous obtiendrez) pour les éditeurs de texte et d'image. Avec GENESIS, un tel slogan devient impraticable. Le son est l'objet *attendu* (ou objet terminal) mais ne peut pas être l'objet d'intérêt pour la manipulation directe.

Au lieu de permettre la manipulation de l'objet terminal (ou de l'objet terminal représenté) on est conduit à proposer la manipulation d'une représentation de l'objet d'intérêt qui par nature, ne peut rendre compte fidèlement de l'objet terminal.

A ces remarques près toutefois, les principes précédents restent valables. Il convient par exemple de focaliser l'attention de l'utilisateur sur la représentation et de rendre compte aussi clairement que possible des actions qu'il entreprend, etc.

Il n'en reste pas moins que dans un modeleur/simulateur tel que GENESIS il ne peut y avoir qu'une corrélation de second ordre entre « ce qu'on voit ou fait », et « ce qu'on obtient ».

Peter Torvik, en interrogeant plusieurs utilisateurs de GENESIS, a observé combien la projection liant la structure et les paramètres d'un modèle CORDIS-ANIMA d'une part et l'objet sonore « terminal » d'autre part est difficile à appréhender par les musiciens [Torvik.01]. L'utilisateur, qu'il soit novice ou expert, éprouve des difficultés à *prévoir* le résultat sonore consécutif à ses actions de modélisations. On pourrait dire, en quelque sorte, qu'au sein de GENESIS WYSMBFWYWH (What You See May be Far From What You Will Hear, ce que vous voyez peut être éloigné de ce que vous allez entendre).

C'est là, nous semble-t-il, une difficulté inhérente à tout processus de modélisation/simulation. Dans GENESIS, c'est qui plus est un principe ou un fondement de l'approche : le décalage entre objet terminal et objet d'intérêt, entre phénomène et

[1] On est ici renvoyé au principe d'économie énoncé plus haut.

cause, est en effet la condition pour qu'une *pensée physique* devienne possible, avec tous les avantages prévisibles que cela comporte (partie I chap. 2) [1].

Face à cette question du *mapping*, il serait tout à fait malvenu de chercher une représentation « phénoménologique » de l'objet physique. L'un des paris – que nous avons cherché à relever est précisément que l'utilisateur puisse progressivement comprendre, apprendre ce *mapping* modèle physique / phénomènes[2].

Ajoutons, enfin, quelques observations à propos des paramètres des modèles. La façon la plus appropriée d'éditer les paramètres des modèles s'avère être, d'expérience, de saisir leur valeur alphanumérique. En conséquence, alors que la représentation se prête bien à l'édition de la structure des modèles, l'édition des paramètres nécessite l'introduction d'objets d'interface. Pour ne pas contredire par trop les recommandations de Beaudouin-Lafon, nous avons essayé que l'utilisateur puisse envisager ces fenêtres d'édition des paramètres non pas comme des objets distincts de la représentation graphique mais comme ses compléments indispensables et naturels. En d'autres termes, nous avons cherché à ce que la représentation graphique et la représentation alphanumérique ne soient pas en concurrence mais se complètent pour donner à (perce)voir, ensemble, l'objet CORDIS-ANIMA.

4.8 - Théorie et principes *vs* pragmatisme et compromis

Nos objectifs en abordant cet état de l'art en matière d'interface homme-ordinateur étaient de résumer le contexte scientifique et les principes opératoires qui ont structuré notre travail de conception de GENESIS. Sans minimiser l'apport de ces connaissances, l'expérience acquise montre que la conception d'une interface comme GENESIS demande avant toute chose une approche pragmatique et un certain art du compromis.

Deux points encore pour soutenir cette opinion.

Le premier. Quand bien même ce travail est le résultat d'une *recherche*, notre objectif était en définitive de permettre à l'utilisateur de créer *effectivement*. L'ensemble de l'environnement dans lequel s'est déroulé le travail a du être considéré, incluant le temps disponible, le coût, les aptitudes des développeurs – dont l'auteur - ainsi que le contexte social et professionnel. Quelle que soit la qualité à laquelle nous voulions parvenir pour GENESIS, ces différents points et la volonté d'arriver à un logiciel « qui tourne » ont imposé, c'est certain, des compromis.

Le second. Il faut se rappeler que théories et principes peuvent conduire à des solutions « théoriquement » valides, mais en pratique irréalisables ou, pire, inutilisables. L'utilisateur, confronté au logiciel n'est guère intéressé par la théorie qui a guidé sa conception. Le pragmatisme consiste alors à se focaliser sur le logiciel tel qu'il est. Il demande que les choix ergonomiques soient évalués non pas seulement au vu des principes et théories mais par les utilisateurs eux-mêmes. Seule l'utilisation effective du logiciel

---

[1] Incidemment, plusieurs des utilisateurs interrogés voient un autre avantage pour la création dans cette propriété : il permet d'obtenir des résultats surprenants, conduisant à ouvrir le projet originellement formulé.

[2] Nous verrons que GENESIS peut offrir, cependant, deux catégories de fonctionnalités qui établissent des ponts entre le modèle et ses propriétés ou les phénomènes qu'il engendre. Ce sont :

- Les outils d'analyse des modèles qui calculent certaines de leurs propriétés et permettent ainsi de *prédire* des caractéristiques des phénomènes générés.
- Les processus de génération qui aident à la conception des modèles en assurant l'obtention de certains phénomènes souhaités.

La partie VIII présentera quelques pistes que nous avons envisagées en la matière.

permet, en dernière analyse, de valider ou d'invalider un choix ergonomique ou une fonctionnalité.

Ainsi, on pourrait dire qu'un principe essentiel pour la conception des interfaces consiste, précisément, à suspecter tout principe et pour s'encrer dans les *faits* et le *faire*. De là, nous sommes conduits à préciser les méthodes et démarches mises en œuvre pour la conception de GENESIS. C'est l'objet du prochain chapitre.

Chapitre 5

METHODE POUR LA REALISATION DE GENESIS

Ce dernier chapitre expose les étapes et aspects méthodiques qui ont marqué notre travail tout au long de la conception de GENESIS. Il présente les phases :

- d'analyse des tâches
- de recherche et de définition des solutions
- d'implémentation
- et enfin d'évaluation.

Pour chacune d'elle, des éléments bibliographiques sont proposés en vis à des conditions de sa mise en œuvre au sein du projet.

Le dernier paragraphe précise la façon dont ces différentes phases se sont articulées. Il montre que la conception de GENESIS a suivi pour l'essentiel un trajet fait d'incrémentations et d'itérations successives.

5.1 - Première étape : Analyse des tâches

> « La première question à se poser, qui est aussi la plus essentielle est : qu'est-ce que l'utilisateur veut faire ? Le processus de conception d'une interface revient encore et encore à cette question fondamentale »[1] [Laurel.90a].

Puisque la première raison d'être d'un système interactif est de permettre la réalisation de tâches par l'interaction de l'homme et de l'ordinateur, il n'est guère étonnant que *l'analyse des tâches* soit très largement discutée dans la littérature. De fait, une bonne démarche pour aborder la conception d'un système interactif consiste à partir de la tâche[2]. Nous précisons, dans ce paragraphe, les méthodes que nous avons adoptées en la matière[3].

5.1.1 - Un processus récursif

Presque tous les problèmes concernant la conception d'une fonctionnalité peuvent être mieux traités si on prend un peu de distance pour se demander ce qu'est la tâche sous jacente. Pour concevoir GENESIS, nous avons ainsi couramment été amenés à confronter les problèmes rencontrés à quelques questions simples, telles :

- Qu'est ce l'utilisateur veut faire, en fin de compte ? (la question de Laurel)

[1] « The first and most important question to ask is :what the user want to do ? The process of interface design (…) returns again and again to this fundamental question ».

[2] Citons par exemple Norman : « nous devrions faciliter la tâche, et non pas travailler l'interface pour la tâche » [Norman.90] et Shneiderman : « tous les *designer* s'accordent à dire que l'ensemble des tâches devrait être déterminé avant que la conception [du système] ne démarre » [Shneiderman.98 p 71] – traductions libres de l'auteur.

[3] Une synthèse des méthodes usuelles pour l'analyse des tâches figure dans [Jeffries.97].

- Comment l'utilisateur souhaiterait-il le faire? Comment lui permettre de le faire ?
- Quelle est la fréquence de la tâche ? (voir le 3ème principe de Hansen, p90).
- Quelle est la complexité de la tâche ?
- Quelles sont les sous-tâches impliquées par la tâche ?
- Etc.

Cette dernière question laisse entendre que l'analyse des tâches est un processus récursif. Shneiderman écrit à ce sujet :

> « Les tâches de haut niveau peuvent être décomposées en plusieurs tâches de niveau intermédiaire, qui à leur tour peuvent être affinées en tâche élémentaires – celles que l'utilisateur exécute par une seule commande, telle la sélection d'une entrée de menu, et ainsi de suite »[1] [Shneiderman.99 p70].

Un exemple nous permettra d'illustrer l'organisation récursive des analyses que nous avons menées.

Nous verrons dans la suite du document que l'utilisation de CORDIS-ANIMA suppose parfois la mise en œuvre simultanée de plusieurs modules. L'analyse de la tâche *travailler de façon macro-modulaire* conduit à introduire certaines fonctionnalités permettant de la soutenir – les notions complémentaires d'Ensemble et de Capsule, que nous introduirons à la partie VI. Appliquée à la notion d'Ensemble, l'analyse fait cette fois apparaître les tâches *définir un Ensemble, nommer un Ensemble, modifier le contenu d'un Ensemble*, etc. Chacune de ces tâches est en outre associée à une certaine fréquence, suivant qu'elle doit être réalisée de façon récurrente ou non. A ce niveau apparaissent d'une part les fonctionnalités nécessaires à la gestion des Ensembles, d'autre part quelques propositions quant à l'ergonomie permettant de les réaliser. On peut alors pister les actions élémentaires que chacune de ces propositions implique et ainsi en faire une évaluation préalable à l'implémentation.

Ainsi, l'analyse des tâches est utile à toutes les étapes de la conception, depuis la description générale de ce qu'il faudra pouvoir faire jusqu'à l'évaluation. A chaque niveau de détail de la conception correspond un degré de précision de l'analyse – et ce le cas échéant jusqu'aux tâches les plus élémentaires, celles par exemple du *keystroke model* basé sur l'approche GOMS[2]. Aussi, quelle que soit la fonctionnalité concernée, le *cahier des charges* qui a accompagné toutes les étapes de sa conception débute, en général, par une analyse des tâches concernées.

5.1.2 - Les processus de création

Au sommet de la pyramide des tâches se trouve l'analyse générale des tâches impliquées par le processus de création avec GENESIS, c'est à dire du processus de création que GENESIS doit permettre et soutenir.

Une première définition de la tâche générale en jeu dans GENESIS pourrait être : « créer des sons et de la musique en utilisant CORDIS-ANIMA et la simulation interactive

[1] « High task actions can be decomposed into multiple middle-level task actions that can be further refined into atomic actions that the user executes with a single command, menu selection, and so on »

[2] Voir également la notion de *micro task analysis* dans [Jeffries.97].

et multisensorielle d'objets physiques »[1]. Elle est cependant bien trop générale pour éclairer d'une quelconque manière la conception du logiciel et une meilleure précision s'impose. Cela, cependant, n'est pas sans poser de difficultés.

C'est que la création semble échapper, par son essence, à la standardisation. Qui plus est, l'approche que nous voulons promouvoir est nouvelle (voir le chapitre 2). Comment, alors, analyser la tâche alors qu'elle n'est pas (encore) observable ? D'une certaine manière, on peut dire que l'étude du processus de création, qui doit être préalable à la conception de GENESIS, suppose en même temps que le logiciel existe – il y a là une contradiction certaine.

Du fait de son importance pour notre propos et de sa difficulté, l'analyse du processus de création avec GENESIS a occupé une place essentielle dans notre travail. La partie III à suivre lui est entièrement consacrée.

5.1.3 - Scénarios

Le recours à des *scénarios* est une méthode originale pour effectuer l'analyse des tâches. Un scénario, c'est selon [Jeffries.97] :

- Une *situation*, c'est à dire une instance concrète d'une tâche replacée dans un contexte plus général ;
- Une *résolution*, c'est à dire une description des étapes qui aboutissent (ou aboutiraient) à la réalisation de la tâche. La résolution peut être effectuée à différents niveaux, suivant le degré de précision souhaitée.

Nous distinguons deux intérêts majeurs pour les scénarios.

Ils facilitent la description d'une tâche

Utilisés pour décrire une tâche *par l'exemple*, les scénarios aident à mieux en appréhender les points clés. Il permettent, en outre, une estimation pertinente des fréquences des sous-tâches. Pour ces raisons, la partie suivante débute par un vaste scénario avec l'exposé de la résolution d'une *situation* très générale : « créer avec CORDIS-ANIMA ».

Ils permettent une évaluation préalable à l'implémentation

Les scénarios sont particulièrement appropriés lorsqu'il s'agit d'évaluer les avantages et inconvénients de plusieurs fonctionnalités ou solutions ergonomiques *envisagées* pour soutenir une tâche.

Nous avons couramment soumis des solutions possibles à des scénarios au cours de la conception de GENESIS. Parmi les scénarios utilisés, on peut citer par exemple :

- « Construire un oscillateur mécanique, le simuler »
- « Construire une corde CA percutée, la simuler, écouter le résultat sonore. Recommencer l'opération pour différents points d'excitation. Sauvegarder le modèle le plus satisfaisant et le son généré »
- ou encore : « Effectuer une démonstration du logiciel ».

[1] En référence à la fonction pédagogique de GENESIS, une tâche secondaire mais du même niveau est : « apprendre CORDIS-ANIMA, les notions essentielles de la physique, et la *pensée physique*».

5.2 - Deuxième étape : Propositions, décision

Lorsqu'une tâche a été identifiée et analysée dans une première version d'un cahier des charges, il devient possible de rechercher des solutions à la fois en termes de fonctionnalité et d'ergonomie.

5.2.1 - Recherche et genèse des idées

Les phases de génération des idées menées au sein du projet GENESIS se caractérisent par une volonté d'ouverture maximale.

- Une coopération étendue

La génération de pistes et nouvelles idées a impliqué plusieurs personnes : Claude Cadoz, l'auteur et l'ensemble des stagiaires qui ont contribué au projet, mais aussi à certaines occassions l'ensemble de l'équipe de l'ACROE et certains utilisateurs.

- Une ouverture d'esprit

Par ailleurs, nous avons cherché à « cultiver notre sensitivité au *design* » ([Norman.90]) et à nous inspirer d'une observation constante du « monde réel » (objets, outils non-informatiques, modes de travail, etc.). Cette ouverture a permis d'identifier quelques métaphores élémentaires pertinentes dans le contexte de GENESIS.

- Ouverture à d'autres logiciels

Nous avons également analysé divers logiciels et cherché à voir dans quelle mesure certaines de leurs caractéristiques pouvaient être transposées dans l'environnement GENESIS. Outre qu'elle aide à la génération d'idées, une telle approche est garante dans une certaine mesure de la *mise en conformité* du logiciel. En plus des logiciels d'usage très courant (bureautique, navigation internet, etc.), nous avons accordé une attention particulière :

- Aux logiciels pour la création musicale[1] - étant entendu qu'on peut espérer une certaine connaissance de leurs principes de la part des utilisateurs de GENESIS ;
- Aux logiciels de CAO (conception assistée par ordinateur), de dessin et d'animation 3D[2] - étant entendu que ces logiciels font souvent appel, comme GENESIS, à une phase de simulation pour l'obtention des résultats et sont porteurs de solutions ergonomiques originales.

On trouvera dans [Mountford.90] quelques autres techniques qui ont pu également nous aider à « générer de nouvelles idées ». Entre autres attitudes, nous avons cherché face à un problème à déterminer si certaines fonctionnalités déjà implémentées pouvaient être étendues ou détournées. Une telle approche tend à éviter un morcellement de l'interface dommageable à son utilisabilité.

Enfin, nous rappellerons qu'à nos yeux l'une des voies les plus efficaces pour progresser consiste, encore et encore, à revenir à la question de Laurel, c'est à dire à se demander *ce que l'utilisateur veut faire* précisément. D'ailleurs c'est souvent lorsqu'on élargit l'angle d'approche qu'apparaît la solution la plus efficace pour la tâche envisagée.

---

[1] Par exemple : MUSIC[N], C-Sound, Max, les logiciels de montage et de traitement, les logiciels de modélisation physique… Notons que [Vinet.99a] et [Torvik.01 pp38-41] proposent une analyse des spécificités des logiciels musicaux.

[2] Par exemple : Blender, Photoshop, etc.

5.2.2 - Retour au cahier des charges, évaluation, décision

Plusieurs pistes ayant été identifiées en termes de fonctionnalité et d'ergonomie, nous en avons sélectionné quelques-unes (typiquement entre 1 et 3) qui paraissaient particulièrement prometteuses afin de les spécifier plus en profondeur. Ces spécifications figurent dans le cahier des charges à la suite de l'analyse des tâches. Elles font appel indifféremment :

- A des descriptions écrites des concepts sous jacents et des éventuelles métaphores ;
- A des esquisses plus ou moins détaillées de l'interface envisagée.
- A des schémas, notamment inspirés des automates à états finis, lorsqu'ils permettent une représentation efficace des algorithmes en jeu.
- Le cas échéant, à une description rapide des structures de données ou des algorithmes impliqués.
- Enfin, dans certains cas (par exemple celui des Ensembles, voir le chapitre 16), nous avons eu ici recours à des *prototypes*.

L'objectif de cette première phase de spécification est d'opérer une évaluation comparée des différentes propositions. Celle-ci a été effectuée en regard :

- De la tâche ou du problème qu'elles sont censées résoudre – il s'agit ici de s'assurer que la solution répond aux besoins de la tâche.
- Des sous-tâches (ou tâches de niveau inférieur) qu'elles induisent. La mise en œuvre de scénarios a été d'une aide précieuse pour les caractériser.
- Des principes exposés au chapitre 4 (« évaluation heuristique », voir aussi le paragraphe 5.4.1 - ci dessous) : S'opposent-elles à la cohérence de l'interface ? Perturbent-elles le modèle mental de l'utilisateur ? Laquelle est la plus *affordante* ? - etc.
- Enfin, de considérations très pragmatiques sur la faisabilité. A ce sujet, rappelons que la conception d'une interface, même au sein d'un projet de recherche, suppose un art certain du compromis.

En règle générale, les résultats de l'évaluation comparée des différentes pistes retenues figurent dans une troisième partie du *cahier des charges*.

5.2.3 - Un cas particulier : l'ergonomie « mise à plat »

Nous évoquons pour terminer ce paragraphe consacré aux spécifications le cas de l'ergonomie « *mise à plat* ».

Pour certaines fonctionnalités critiques ou d'approche difficile, nous avons décidé d'implémenter une ergonomie sans relief, sans qu'une solution optimale ne soit dans un premier temps recherchée. Ce faisant, nous avons pu confronter la fonctionnalité à l'utilisateur et ainsi mettre en œuvre une analyse critique d'un existant.

Cette analyse a permis dans un premier temps :

- D'observer l'ensemble des tâches induites par la fonctionnalité, d'identifier les tâches difficiles et récurrentes. De mieux comprendre les problèmes clés sous-tendus par la fonctionnalité.
- De distinguer les aspects positifs de l'ergonomie « à plat » (quand bien même, dans l'ensemble, elle n'est guère satisfaisante) et, au contraire, ceux qui posent problème.

Dans un second temps, nous avons pu reprendre le problème à la base et entreprendre une révision profonde de l'ergonomie.

On trouvera quelques exemples de cette démarche dans la partie VII consacrée à l'ergonomie des fonctionnalités de simulation. Dans l'ensemble, nous considérons qu'elle a apporté des résultats satisfaisants, et ce malgré son coût en termes de développement et les difficultés qu'elle a pu entraîner auprès des utilisateurs de la version intermédiaire.

5.3 - Troisième étape : Implémentation ; génie logiciel

Dans l'étape d'implémentation on trouve d'une part le développement des structures de données et des fonctions relatives à sa gestion (notamment la couche d'abstraction), d'autre part le développement de l'interface utilisateur. Incidemment, l'expérience a confirmé que le second est de loin le plus long et le plus difficile.

Ce paragraphe précise quelques points de méthode observés pour l'implémentation.

5.3.1 - Spécifications techniques

Sur la base des spécifications externes de la solution retenue, l'analyse technique et les spécifications techniques se sont appuyées : (1) sur les conventions de codage ; (2) sur une connaissance des architectures théoriques des systèmes interactifs ; (3) sur une connaissance du code source déjà existant.

Cette phase d'analyse et de spécification a entre autre permis d'identifier les principales causes d'utilisation des ressources de la machine et, le cas échéant, d'envisager des solutions d'optimisation.

5.3.2 - Développement parallèle et intégration

Le développement de GENESIS résulte, dans des proportions modestes, d'un travail d'équipe : plusieurs stagiaires ont rejoint tour à tour le projet pour des périodes plus ou moins longues. Une stratégie de développement en équipe était donc nécessaire. Elle s'organise ainsi :

- Un compte est dédié aux sources des versions officielles de GENESIS.
- Chaque développeur copie la version courante du logiciel au début de son travail. Il est responsable de son propre versionnage. Le développement est donc effectué en parallèle.
- Chaque développeur est responsable de la tenue d'un *cahier de modification* qui regroupe au jour le jour les principales modifications et ajouts au code.
- Lorsque la fonctionnalité développée a été soumise à une première série de tests par le développeur et par les responsables du projet, l'intégration du travail au sein des versions officielles est réalisée.
- A l'issue de l'intégration, un rapport technique réorganise l'ensemble du cahier de modifications.

5.3.3 - Le versionnage de GENESIS

Ce paragraphe précise le processus et les conventions adoptés pour le versionnage des sources.

De façon générale, une nouvelle version de GENESIS a été générée à l'issue de chaque phase d'intégration.

Une version de GENESIS est repérée par un code à trois chiffres et une lettre minuscule – par exemple 1.42a. Le premier chiffre est resté inchangé tout au long de notre

travail, qui porte donc sur la version 1 de GENESIS. Les trois autres codes permettent trois niveaux de versionnage, comme le montre le tableau suivant.

<table>
<tr><td rowspan="2">Niveau 1 : Chiffre principal</td><td>1.[123…]</td><td>Le niveau 1 est incrémenté dès lors que des fonctionnalités majeures sont ajoutées au logiciel ou que le code est modifié en profondeur. Le premier niveau correspond à un cycle approximatif d’un an.</td></tr>
<tr><td colspan="2">La version 1.3 correspond à la refonte complète de la structure de données et de l’architecture du programme, à une redéfinition des fenêtres de paramètres et conditions initiales et à l’introduction du principe de la sélection.
La version 1.4 correspond à une première refonte de l’ergonomie des simulations et à l’introduction de nombreuses fonctionnalités (analyse, multiplication des paramètres, substitution par exemple).
La version 1.5 (version courante) correspond à l’introduction des fonctionnalités de macro-modularité, à la refonte des fonctionnalités de navigation et de manipulation directe, à l’introduction du système de série de structures et à une nouvelle version de l’ergonomie des fonctionnalités de simulation.</td></tr>
<tr><td>Niveau 2 : Chiffre secondaire</td><td>1.n[123…]</td><td>Ce second niveau correspond à des modifications plus ou moins importantes des principes déjà en place, par exemple des changements d’aspect de l’interface ou des ajouts de raccourcis. Sa durée de vie varie entre 1 mois et 6 mois.</td></tr>
<tr><td>Niveau 3 : Lettre code</td><td>1.nm[abc…]</td><td>Le troisième niveau correspond à la correction de bugs et aux cas ou un comportement légèrement en désaccord avec les spécifications initiales est observé. La durée de vie d’une version de troisième niveau peut descendre jusqu’à une semaine.</td></tr>
</table>

Figure 17 : processus et convention de versionnage de GENESIS

Ajoutons enfin quelques mots sur les versions ad’hoc. Au cours de notre travail, certains utilisateurs (compositeurs et chercheurs) ont fait état de besoins spécifiques dont la satisfaction nécessitait la conception de fonctionnalités dédiées. Ces versions ad hoc n’ont, en général, pas donné lieu à une recherche poussée en matière d’ergonomie, l’essentiel étant de répondre ponctuellement à une demande particulière. Elles ont par contre permis une *évaluation* de la fonctionnalité implémentée, conduisant le cas échéant à sa généralisation. Les versions ad hoc du logiciel sont archivées, comme les autres, sur le compte de développement. Parmi les versions ad hoc, citons par exemple :

- *G1.32 NLX3 - 04-00* : implémentation d’une raideur non linéaire en cube de la position des MAT connectés (voir chapitre 10 et l’Annexe B).
- *G1.40b GesteFourcade - 07-00* : implémentation de deux flux d’entrée dynamique (position et force) pour disposer d’un *mode Jeu* élémentaire.
- *G1.42a MesureLNL - 06-01* : implémentation de fonctionnalités de métrologie pour les liaisons non-linéaires LNL (étude du frottement d’archet en temps différé).
- *G1.42b TempsRéel - 08-01* : connexion au moteur temps réel pour expérimentation.

5.4 - Evaluation et tests

L'évaluation des systèmes interactifs est multi-dimensionnelle. Pour évaluer un système, il convient de considérer notamment sa fiabilité, son utilité, son utilisabilité[1], l'efficacité qu'il permet, la facilité d'apprentissage ou d'auto-apprentissage et son agrément (la liste n'est pas exhaustive).

La plupart de ces dimensions sont difficiles à envisager sur la base de critères objectifs. L'évaluation, en conséquence, est par nature sujette à controverses. Par ailleurs, aucune méthode ne pouvant couvrir tous ces aspects, il convient pour évaluer un système d'avoir recours à plusieurs *méthodes et techniques* complémentaires [Baecker&al.95c], [Shneiderman.98 p95-118], [Johnson.92 pp84-99], [Usabilis].

Le classement des méthodes d'évaluation que propose Naël (figure 18) montre que l'évaluation concerne *toutes* les étapes du processus de conception, depuis les toutes premières spécifications sur papier jusqu'aux versions diffusées (on parle alors de *tests d'acceptabilité)* en passant par les versions intermédiaires du développement. Tous les auteurs se rejoignent pour dire qu'il est *nécessaire* d'intégrer l'évaluation au cœur même de la conception d'un système interactif. En accord avec ce conseil, *l'évaluation aura été une préoccupation permanente tout au long de notre travail.*

	Données subjectives	Données objectives
IHM représentée (système en projet)	Avis d'experts	Méthodes analytiques
IHM réelle	Avis des utilisateurs	Méthodes d'observation

Figure 18 : Plusieurs catégories de méthodes pour l'évaluation
d'après [Naël.96]. Les méthodes grisées figurent parmi celles que nous avons mises en œuvre pour l'évaluation de GENESIS

Les paragraphes précédents ont évoqué quelques-unes des démarches que nous avons utilisées pour évaluer des propositions avant de leur développement (« IHM représentée »). Ce paragraphe présente plus spécifiquement les voies empruntées pour l'évaluation des différentes versions du logiciel (« IHM réelle »).

5.4.1 - Méthodes analytiques

Les méthodes analytiques consistent à évaluer le système sans avoir recours à une campagne de tests. Elles sont donc praticables sur « IHM représentée » mais demeurent très utiles pour évaluer une version développée. Elles méritent ici quelques précisions[2].

On peut distinguer les méthodes analytiques objectives (i.e. : qui offrent des résultats objectifs) et les méthodes analytiques subjectives.

[1] Rappelons e que pour certains auteurs la notion d'utilisabilité prend en compte plusieurs des dimensions ici proposées. Les sites Internet [Utilsab.] et [Usabilis] proposent un exposé clair et détaillé des multiples significations de la notion d'utilisabilité.

[2] Il est généralement conseillé de faire appel à un expert indépendant pour les mettre en œuvre. Ce n'est pas, cependant, ce qui a été fait pour GENESIS – voir à ce sujet le paragraphe 5.4.4 - p116.

Les méthodes objectives reposent essentiellement sur la résolution de scénarios. L'approche GOMS (paragraphe 3.2.2 -71) en est un exemple : elle conduit, en théorie, à une estimation du temps nécessaire à la réalisation de tâches élémentaires. La méthode dite d'exploration cognitive (*cognitive walkthrough*) n'en est guère éloignée, quoi qu'elle soit plus directement dédiée au cas des utilisateurs novices et à la phase d'apprentissage[1]. Elle consiste à pister pas à pas toutes les opérations nécessaires à la réalisation d'une tâche donnée, frappe de touche et déplacement souris compris, à noter ces étapes successives puis à analyser les résultas obtenus – en se demandant notamment si, à chaque instant, le système invite l'utilisateur à réaliser l'étape suivante.

Nous n'avons pas mis en œuvre les techniques d'évaluation objective de façon systématique ; le type d'analyse proposé par l'approche GOMS (décomposition en objectif, méthode(s), opérations et règles de sélection) et l'exploration cognitive auront cependant largement inspiré plusieurs phases d'analyse de GENESIS.

Les méthodes analytiques subjectives, aussi appelées « évaluations heuristiques » consistent essentiellement à confronter le système à un ensemble de critères ou *check-list*. On trouvera dans [Baecker&al.95c] un résumé des critères de Nielsen dont nous nous sommes inspirés. La majorité d'entre eux constituent en fait une reformulation sous forme interrogative des principes que nous avons exposés au chapitre précédent. Pour évaluer un système, il s'agit par exemple de se demander si l'interface est *cohérente* et *conforme* ; si elle fait apparaître l'état du système ; si les opérations disponibles sont visibles ; si le système est économe en fenêtres, *widgets* ou couleurs, etc.

La portée de l'évaluation heuristique est par nature limitée (voir par exemple [Shneiderman.98 p126-151]) : les critères sur lesquels elle se base sont souvent vagues, parfois contradictoires et dans tous les cas soumis à l'interprétation de l'évaluateur. L'évaluation heuristique est en revanche particulièrement aisée à mettre en œuvre de façon informelle ; nous l'aurons donc pratiquée continuellement, pendant les phases d'analyse des tâches, de génération d'idées, de spécification et de développement comme pour l'évaluation terminale des versions finalisées.

5.4.2 - Tests par les développeurs et « *essayez de tout casser !* »

Des tests continus ont été pratiqués pendant le développement essentiellement pour valider les différents modules développés ou modifiés et pour mettre au jour des bugs et problèmes d'ordre technique qui n'avaient pas été prévus.

Les tests « essayer de tout casser » (*try-to-destroy-it*) constituent une manière efficace d'évaluer la fiabilité d'un système[2]. Au sein du projet GENESIS, ces tests ont été mis en œuvre à l'issue de chaque phase d'intégration. Ils ont été en général organisés en « compétition » avec plusieurs employés ou stagiaires non directement liés au projet, auxquels il était demandé de faire apparaître un « bug » ou défaut majeur le plus rapidement possible en sollicitant l'interface de toutes les façons imaginables. Ils ont largement contribué à la stabilité de GENESIS – point évidemment nécessaire pour qu'une relation de confiance puisse s'installer entre l'utilisateur et le programme.

[1] Conçus par Polson initialement pour évaluer les systèmes appelés à être appris au fur et à mesure de leur utilisation, l'exploration cognitive (*cognitive walkthrough*) a montré un intérêt plus général. Voir [Shneiderman.98 p126], [Baecker&al.95c] par exemple.

[2] Ils ont à l'origine été mis en œuvre pour les jeux vidéos. Shneiderman insiste sur leur importance, et préconise que tout projet y fasse appel [Shneiderman.98].

5.4.3 - Confrontation aux utilisateurs

Nous considérons que les méthodes analytiques ne permettent pas d'obtenir un panorama complet de la qualité d'une interface – même mises en œuvre par des experts. Par ailleurs, il est certain que les concepteurs ne peuvent pas avoir une vision objective du système lorsqu'ils l'utilisent ; ils ne peuvent le tester de façon valable dans son intégralité[1].

Aussi, en matière d'évaluation, *rien ne remplace en définitive la confrontation du système aux utilisateurs dans une situation d'utilisation réelle* (ou *in situ*), c'est à dire pour GENESIS dans une situation de création.

L'évaluation impliquant des utilisateurs peut être de deux natures :

- Les tests d'utilisabilité (*usability testing*), qui consistent à *observer* le comportement effectif d'utilisateurs face au système, dans des conditions expérimentales (i.e. : en laboratoire) ou sur le terrain.
- Les enquêtes, qui consistent à recueillir par divers procédés les avis, critiques ou suggestions des utilisateurs relativement au système.

Pour ces deux types de démarches, le recours à une méthodologie soigneusement conçue garantit, en théorie, que soient évités certains travers ou dérives – par exemple une trop grande proximité entre l'utilisateur-testeur et l'expérimentateur, etc.

En ce qui concerne la méthodologie des enquêtes, nous signalons les travaux de Shneiderman [Shneiderman.98 pp132-143]. On y trouvera notamment quelques exemples de questionnaires oraux ou écrits.

En ce qui concerne les tests d'utilisabilité, Naël précise qu'un panel de dix testeurs est suffisant (au-delà, on ne recueille plus guère de nouvelles observations) et offre une synthèse des recommandations courantes qu'on peut trouver dans la littérature [Naël.96][2]. Il suggère notamment :

- Que les conditions expérimentales soient clairement annoncées : tâches ou scénarios demandées aux testeurs, durée du test, enregistrement vidéo ou non, etc.
- Que les tâches choisies soient représentatives des conditions réelles d'utilisation.
- Que le panel de testeurs soit représentatif des utilisateurs potentiels *et de leur diversité* (âge, connaissance des environnements informatiques et du domaine, catégorie socio-professionnelle, etc.).
- Que la présentation du logiciel effectuée avant le test soit conforme à celle qui sera proposée finalement (guide d'utilisation, démonstration, etc.).
- Que l'expérimentateur n'intervienne pas pendant le test.

[1] Cela reste vrai lorsque le concepteur est aussi utilisateur – cas, pour GENESIS, de Claude Cadoz. La volonté « de se mettre dans la peau d'un utilisateur » est bien sûr nécessaire tant pendant la phase de conception que pour l'évaluation ; elle reste, cependant, en grande partie une illusion.

[2] Nous renvoyons également le lecteurs aux conseils de Nielsen (voir [Baecker&al.95c]) et de Gomoll [Gomoll.90]. Outre qu'elle propose 10 étapes particulièrement claires pour la mise en œuvre des tests d'utilisabilité, Gomoll rappelle deux techniques qui ont largement prouvé leur intérêt : demander aux testeurs d'expliquer continuellement à haute voie ce qu'ils font ou cherchent à faire, et filmer les séances afin qu'une analyse plus complète puisse en être faite ultérieurement.

Ces importantes recommandations sont évoquées ici à titre d'information ; *elles n'ont été que très partiellement respectées au sein du projet*. Par manque de temps et de savoir-faire notamment, les confrontations de GENESIS aux utilisateurs n'ont en effet pas donné lieu à la mise en œuvre d'une méthodologie strictement cadrée.

Pour autant, les versions successives de GENESIS ont *toutes* été effectivement utilisées dans des perspectives de création, de recherche *et* de pédagogie, c'est à dire dans des situations exigeantes et par des utilisateurs terminaux. Nous avons donc été *tout au long de notre travail* en contact avec des utilisateurs, et nous avons très souvent eu recours à leur observation et bénéficié de retours, d'avis et de suggestions de leur part. Sans prétendre que cette situation remplace une phase de test ou d'enquête plus formalisée, elle a très largement participé à l'évaluation de GENESIS et ainsi à sa conception.

Utilisation au sein du laboratoire

Chaque version de niveau 3 (1.nm[abc…]) ayant passé les tests « *essayez de tout casser* » a été systématiquement installée sur le réseau de calcul du laboratoire.

En règle générale, 1 à 5 utilisateur(s) travaille(nt) avec GENESIS au sein du laboratoire : compositeurs en résidence, physiciens, acousticiens, etc. Leur présence a grandement contribué au développement de GENESIS en permettant un cycle conception/évaluation/modification particulièrement court.

Par ailleurs, chaque stagiaire appelé à rejoindre le projet a débuté ses travaux par une phase de confrontation au logiciel, à l'issue de laquelle un court rapport a été produit. Le travail de Peter Torvik (stage de DEA ATIAM[1]), propose notamment une analyse de la version 1.3 de GENESIS, éclairée de nombreuses remarques d'utilisateurs [Torvik.01].

GENESIS n'est pas seulement utilisé au sein du laboratoire. Depuis le début de nos travaux, sa diffusion s'est organisée et a constamment augmenté. On peut distinguer plusieurs types d'utilisateurs, plusieurs catégories d'objectifs et en conséquence plusieurs démarches d'évaluation. Nous les précisons dans la suite.

Utilisateurs partenaires du projet

Giuseppe Gavazza, Hans-Peter Stubbe, Ludger Brummer et Mesias Maiguashka ont de longue date manifesté leur intérêt pour les travaux de l'ACROE en matière de modélisation physique. Depuis plusieurs années, ils utilisent intensément GENESIS à des fins de création musicale[2]. Ils sont à ce jour, aux côtés de Claude Cadoz, les utilisateurs les plus exigeants vis à vis du logiciel, les plus demandeurs quant à ses fonctionnalités, les plus attentifs à son efficacité. Ils sont, enfin, compréhensifs quant au fait que GENESIS est un environnement expérimental.

Ces compositeurs sont ainsi devenus des partenaires privilégiés du projet[3] – qu'ils soient, ici, remerciés de leur disponibilité.

---

[1] Acoustique, Traitement du signal et Informatique Appliqués à la Musique – Paris VI, Aix-Marseille II, INPG Grenoble.

[2] Citons quels œuvres : *Rhizome* (Hans-Peter Stubbe), *La Natura della cose* (Giuseppe Gavazza), *Tiefen* (Mesias Maiguashka), *Le temps s'ouvre* (Ludger Brummer).

[3] Notons qu'on s'approche ainsi de la méthode dite de la *conception participative* (*participatory design*) [Olsen.97], même si ils n'ont pas été systématiquement impliqués dans la conception de GENESIS.

L'interaction avec ces compositeurs a eu lieu par échange de mails et au cours de rencontres informelles[1]. Leur importante contribution à nos travaux peut s'analyser en trois points :

- Ils ont, tout d'abord, expliqué leur processus de création et la place qu'y tient GENESIS, nourrissant ainsi l'analyse des tâches que nous avons effectuée.
- Ils ont, à notre demande, exprimé leurs opinions sur les versions successives du logiciel, indiquant notamment leur satisfaction et leur frustration et participant ainsi activement à l'évaluation (évaluation subjective).
- Enfin, ils ont proposés des pistes pour le développement de nouvelles fonctionnalités, en indiquant notamment ce qu'ils *aimeraient* pouvoir faire avec GENESIS (nouvelle tâche). Si plusieurs suggestions ont inspiré nos travaux, notons cependant que nous ne les avons pas systématiquement considérées. Certaines, en effet, divergeaient par trop de notre ordre de priorités. D'autres s'inspiraient de démarches traditionnelles et semblaient en désaccord avec notre objectif de promouvoir un nouveau processus de création basé sur une « pensée physique »[2].

Diffusion dans les centres de création

GENESIS est mis à disposition depuis plusieurs années d'étudiants, enseignants et artistes dans plusieurs centres de création[3] et de formation. Il est en particulier au cœur d'une classe de composition, encadrée par le compositeur Arnaud Petit, au sein du Conservatoire National de Région de Grenoble.

Cette catégorie de diffusion du logiciel appelle deux remarques, qui illustrent en quoi les données recueillies étaient spécifiques et utiles :

- Tout d'abord, elle s'accompagne d'un encadrement pédagogique, et notamment de séances de formation assurées par Claude Cadoz. En conséquence, les utilisateurs qui ont par ce biais accès au logiciel peuvent être régulièrement suivis et *observés*.
- Ensuite, elle s'adresse à des utilisateurs à la fois novices et motivés, caractéristique intéressante pour l'évaluation puisqu'elle permet d'observer sur le long terme la prise en main du logiciel.

Workshops

Régulièrement, le laboratoire organise des workshops[4] autour des logiciels développés, permettant aux participants de se confronter aux principes et méthodes originaux qu'ils sous tendent sur une période courte et de manière encadrée. Ces workshops concernent indifféremment des groupes ou des participants isolés. Ils réunissent des personnes d'horizons très divers, qui toutes sont motivées par un but spécifique. S'y inscrivent aussi bien des pédagogues intéressés par GENESIS dans le cadre d'enseignements, des chercheurs en Informatique Musicale, des compositeurs désireux de découvrir de nouvelles approches, etc.

---

[1] On trouvera en matière d'exemple certains mails à l'annexe D.

[2] A plusieurs reprises, par exemple, il nous a été demandé qu'un procédé permettant de contrôler les modèles par des *signaux* soit implémenté ; c'est là une étape qu'il faudra probablement franchir tôt ou tard, mais elle n'est pas prioritaire. Plus, elle demande qu'une réflexion approfondie soit conduite sans quoi GENESIS risque de trahir notre volonté de baser le processus de création sur la « pensée physique », évidemment très éloignée de la manipulation de signaux.

[3] Depuis 1996 au ZKM de Karlsruhe, à l'AIAM de Turin, etc.

[4] Citons. l'école d'art d'Angoulême en Mars 2002 ou les Rencontres Internationales Informatique et Création Artistique (Novembre 2001).

Cette diversité des participants, de leur expérience et de leurs objectifs respectifs, alliée à quelques considérations plus pragmatiques (réunion de plusieurs utilisateurs dans un même espace sur plusieurs jours successifs), confère au workshops un statut à part pour l'évaluation. Deux méthodes d'évaluation ont été pratiquées :

- Evaluation subjective : avis, remarques, suggestions

Lors des workshops, chaque participant s'est vu confier un cahier de remarques sur lequel il était invité à s'exprimer. Un recueil oral – ensuite traduit sur papier – des avis, remarques et suggestions a de plus été effectué.

- Evaluation objective : observation in-situ

Lors des workshops, nous avons conservé un œil attentif sur les stratégies des utilisateurs, leurs méthodes de découverte des fonctionnalités et les différents problèmes qu'ils ont rencontrés. Ces observations ont été consignées méthodiquement et utilisées pour les développements en cours.

5.4.4 - Conclusion sur l'évaluation

- Par manque de temps,
- Par manque de savoir-faire en la matière,
- Et surtout parce qu'à nos yeux les versions 1.3 à 1.42b n'étaient pas suffisamment finalisées pour justifier la mise en œuvre d'un processus d'évaluation lourd,

nous n'avons pas eu recours, au cours de nos travaux, aux méthodes d'évaluation telles qu'elles sont formalisées dans la littérature.

Nous nous sommes en revanche largement inspirés des méthodes proposées dans la littérature pour conduire des évaluations informelles (analytique ou via des utilisateurs). Plusieurs documents rendent compte de ces évaluations continues (certains sont proposés à titre d'exemple à l'annexe D).

La version 1.5 de GENESIS aujourd'hui disponible nous semble suffisamment aboutie pour qu'un test d'utilisabilité puisse être entrepris en bonne et due forme dans un proche avenir.

De façon générale, l'évaluation a été intégrée au cœur même de la méthodologie adoptée pour la conception de GENESIS. Elle n'a pas été envisagée comme une étape terminale qui viendrait clôturer un travail, mais comme une démarche active qui participe à la conception elle-même.

Ce dernier point nous conduit naturellement au paragraphe suivant, qui va porter un regard plus général sur la façon dont GENESIS a été développé et terminer ainsi cette partie.

5.5 - Démarche incrémentale et itérative ; les boucles imbriquées

L'organisation de ce chapitre peut laisser croire qu'une méthodologie traditionnelle proche des modèles en V ou Top / Down a été mis en œuvre pour le projet GENESIS, enchaînant cahier des charges / spécifications générales / spécifications techniques / implémentation / tests de conformité et enfin évaluation. C'est effectivement le cas pour les cycles de quelques semaines ou mois qui ont conduit à l'implémentation de nouvelles fonctionnalités. Mais, en fait, la progression d'ensemble a été dans son ensemble fondamentalement différente. La façon la plus juste de la qualifier serait de dire qu'elle a suivi un trajet fait d'incrémentations et d'itérations successives.

5.5.1 - Démarche incrémentale

Nous avons dès le début de nos travaux cherché à établir un panorama complet des tâches impliquées dans l'utilisation de CORDIS-ANIMA et de la simulation interactive et multisensorielle d'objets physiques, c'est à dire des grandes lignes des processus de création que devait soutenir GENESIS (voir la partie suivante). Par contre, nous avons envisagé les fonctionnalités nécessaires les unes après les autres, partant des plus fondamentales pour aller vers les plus optionnelles.

Une telle *démarche incrémentale* s'accompagne d'une difficulté certaine pour l'implémentation, puisqu'elle interdit par nature que l'ensemble des fonctionnalités soit connu à l'avance et puisse guider la conception de l'architecture du programme.

Elle nous a semblé, en revanche, toute indiquée dans le contexte de GENESIS. Elle a permis *d'une part* que la diffusion du logiciel et les expérimentations et créations soient menées sur des bases certes incomplètes mais saines. Elle a permis d'autre part la conduite d'une réflexion progressive sur des fonctionnalités de plus en plus complexes, nourrie à la lumière d'une analyse des tâches sur les versions successives.

GENESIS s'est donc enrichi progressivement au cours de nos travaux.

A ce jour, certaines fonctionnalités sont prévues et spécifiées, mais pas encore implémentées dans la version diffusée de GENESIS. Ces fonctionnalités seront décrites évoquées dans les différents chapitres consacrés aux perspectives. D'autres, en revanche, sont encore au stade embryonnaire. La *démarche incrémentale* devrait, à l'avenir, permettre de les aborder.

5.5.2 - Démarche itérative, boucles imbriquées

> *«* La seule approche possible [pour la conception d'une interface] est une approche itérative selon laquelle des prototypes sont réalisés puis évalués afin de valider, remettre en cause ou affiner les choix de conception *» [Beaudouin.99].*

Tous les textes traitant du processus de conception des systèmes interactifs insistent, à la manière de Beaudouin-Lafon, sur la nécessité absolue d'une *démarche itérative*.

Conformément à ce conseil, la méthodologie mise en œuvre pour GENESIS fait apparaître des boucles imbriquées (figure 19 page 114). Chaque boucle a sa propre constante de temps (de quelques jours à un an) et comprend, à minima, une phase préalable

d'analyse des tâches, une phase de conception, une phase d'implémentation et une phase d'évaluation.

La démarche itérative appelle plusieurs remarques :

- C'est la seule qui permette d'intégrer l'évaluation au cœur de la conception. Une telle évaluation en ligne est alors moins une estimation de la qualité du travail accompli qu'une base pour les développements futurs.
- Elle promeut le recours à des prototypes de différents niveaux de finalisation.
- Elle impose que des fonctionnalités ou solutions ergonomiques soient abandonnées sans regret lorsqu'elles s'avèrent présenter des défauts majeurs, et ce quels que soient les efforts fournis pour leur implémentation.
- Elle suppose, enfin, que le passage des spécifications théoriques à l'implémentation soit effectué au bon moment. Il faut bien sûr spécifier toute solution implémentée mais à l'inverse il faut savoir limiter les spécifications puisqu'elles ne peuvent que *partiellement* prendre en compte les multiples enjeux du système.

Les démarches incrémentales et itératives constituent donc les caractéristiques fondamentales de la méthodologie que nous avons adoptée. Elles se placent, incidemment, dans la droite ligne du pragmatisme auquel nous appelions à la fin du chapitre précédent : avec elles, toute proposition se doit d'être mise à l'épreuve quelle que soit son apparente limpidité, et ne sera validée qu'après avoir été confrontée à la pratique.

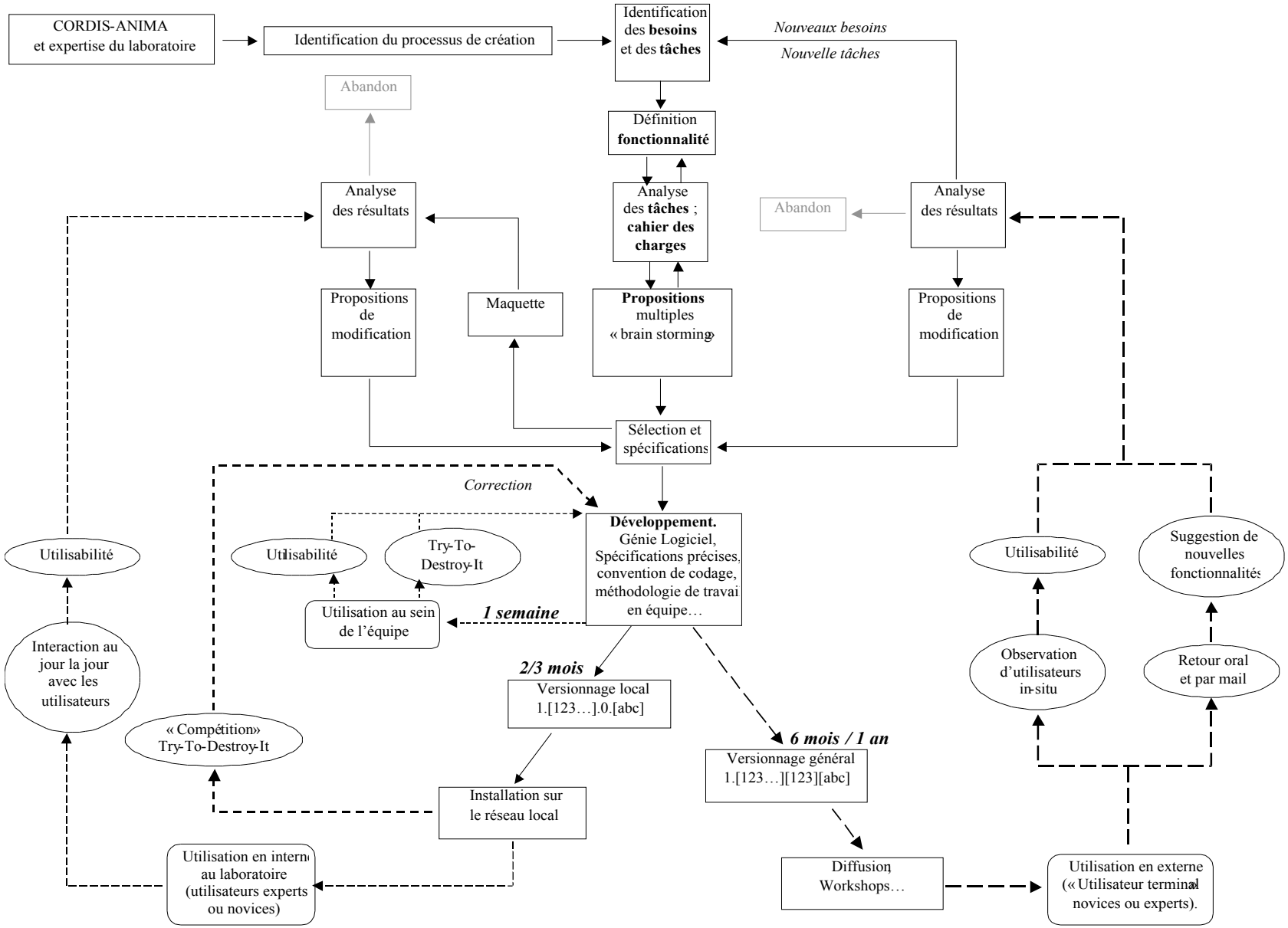


Figure 19: les boucles imbriquées de la méthodologie itérative adoptée pour la conception de GENESIS

CONCLUSION

Dans cette partie nous avons :

- Donné un aperçu des théories et modèles relatifs à la psychologie humaine qui ont constitué un cadre théorique pour la conception de GENESIS ;
- Proposé un bref parcours de l'état de l'art en matière d'interface et d'architecture logicielle ;
- Exposé les principaux principes et guides inspirés de la littérature que nous avons considéré pour concevoir GENESIS ;
- Présenté la méthodologie mise en œuvre pour la conception des fonctionnalités et de l'ergonomie de GENESIS.

Nous sommes maintenant à même de nous consacrer à la mise en œuvre de cette méthodologie, c'est à dire de nous consacrer à l'environnement GENESIS proprement dit.

La partie III est consacrée à l'*analyse générale des tâches*. Les principaux aspects du *processus de création* que nous entendons soutenir avec GENESIS y seront exposés. La partie III constitue ainsi une base pour l'ensemble des parties suivantes.

La suite du document présente la version 1.5 actuellement disponible. Plus généralement elle montre ce que devraient être, selon nous et compte tenu des différents enjeux, une version optimale de GENESIS.

La partie IV présente les *choix* effectués au niveau de l'organisation générale de GENESIS. Elle présente et justifie les quelques restrictions essentielles qu'il nous a fallu faire pour mener à bien nos travaux. Elle expose l'organisation générale du logiciel, puis les principaux choix de nature ergonomique qui se diffusent, dans suite, au niveau de toutes les fonctionnalités.

La partie V est consacrée aux fonctionnalités fondamentales que sont la représentation des objets CORDIS-ANIMA, la manipulation directe du réseau et les fenêtres d'édition des paramètres. La partie VI considère, alors, les fonctionnalités complémentaires et évoluées d'édition.

La partie VI, enfin, présente les fonctionnalités de simulation telles qu'elles apparaissent dans la dernière version de GENESIS.

Partie III

Le processus de création avec GENESIS

Analyse des Tâches

> Au delà des problèmes concrets de mise en œuvre, la principale difficulté pour approcher cette question [concevoir une interface pour la création musicale] tient d'abord à l'impossibilité de définir complètement et objectivement les fonctions d'un outil de création. (…) Les témoignages de compositeurs concordent sur l'importance, dans le processus de création, de décisions qui échappent à toute rationalisation immédiate. (…) il existe une contradiction fondamentale entre singularité et régularité, entre la démarche du compositeur, qui tend à affirmer son originalité et sa différentiation et le développement logiciel, qui vise la mise à disposition d'un même ensemble de fonctions auprès d'une communauté d'utilisateurs.
>
> Hugues Vinet
> [Vinet.99b]

Quels peuvent-être les objectifs d'un utilisateur relativement à la simulation interactive et multisensorielle d'objets physiques ? Que doit-il et que va-t-il faire pour les atteindre ? Quelles démarches et quelles stratégies va-t-il adopter ? Quelles difficultés va-t-il rencontrer ? En d'autres termes, quel(s) processus de création devons-nous soutenir avec GENESIS ?

« L'analyse des tâches » doit être préalable à la conception de GENESIS afin que notre travail soit fondé sur les nécessités de la création. Pourtant, du fait même de la nouveauté de la démarche que nous voulons promouvoir, ce n'est qu'à travers l'observation effective du dialogue utilisateur/logiciel qu'elle pourrait être élaborée sûrement. Le contournement de ce paradoxe a été, de fait, l'un des objets de nos recherches. Pour l'effectuer, nous nous sommes appuyés sur plusieurs considérations (figure 1).

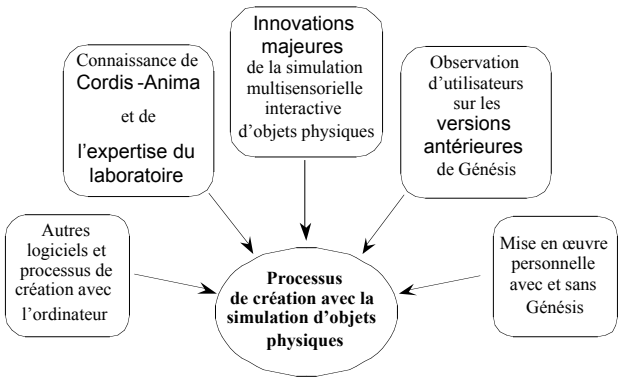


Figure 1 :
Méthodologie pour l'analyse des tâches

- Une connaissance des différents éléments de l'outil, particulièrement de CORDIS-ANIMA ainsi que de l'expertise du laboratoire.
- L'identification des innovations que propose le paradigme de la simulation interactive et multi-sensorielle d'objets physiques pour les démarches de la création musicale (partie I chapitre 2) – le processus de création repose en profondeur sur ces innovations.
- L'observation de compositeurs en situation de création sur les versions successives du logiciel et l'écoute attentive de leurs remarques.

Et enfin, dans une moindre mesure :

- Une confrontation personnelle à l'outil à l'aide d'un environnement élémentaire de programmation et des moteurs de simulation – voir par exemple l'annexe B.
- L'étude d'autres logiciels de modélisation physique et de création musicale, pour déterminer leurs fonctionnalités communes et les processus traditionnels de la création avec ordinateur qui, pour certains, ont pu être retrouvés dans GENESIS.

Le chapitre 6 propose un *scénario* : partant d'un objet très simple pour aboutir à une structure complexe, il introduit par l'exemple quelques-uns des processus essentiels de la création avec CORDIS-ANIMA.

Les chapitres 7, 8 et 9 synthétisent cette première approche. Ils proposent une présentation précise et organisée des différentes tâches que GENESIS doit soutenir et constituent ainsi une base pour les parties suivantes.

Introduction

Contexte et bibliographie

Partie I : le paradigme de la simulation interactive et multisensorielle d'objets physiques

Partie II : Interaction Homme-Machine

Partie III : Le processus de création avec GENESIS. Analyse des Tâches

Chapitre 6 Un parcours simple pour la création 126

6.1 - L'OBJET LE PLUS SIMPLE 126
6.2 - STRUCTURES VIBRANTES 129
6.3 - COM-POSITION DE STRUCTURES VIBRANTES. CHEVALET, ENVIRONNEMENT LOCAL, ENVIRONNEMENT GLOBAL. 132
6.4 - EXCITATEUR, MODIFICATEUR, DÉCLENCHEUR 135
6.5 - INSTRUMENTISTE, CHEF, DÉCLENCHEURS 137
6.6 - FIN DE LA PHASE DE CONCEPTION 139
6.7 - JEU ET INTERACTION MULTISENSORIELLE 140
6.8 - L'OBJET " TRACE " ET L'INCRÉMENTATION DU JEU 140
6.9 - EDITION DE LA TRACE 141

Chapitre 7 Considérations générales sur la création avec CORDIS-ANIMA 143

7.1 - GENESIS, UN ENVIRONNEMENT COMPLET 143
7.2 - SITUATION TEMPS-RÉEL ET TEMPS-DIFFÉRÉ ; LA SIMULATION D'OBJETS PHYSIQUES 144
7.3 - COMPOSANTES STRUCTURELLE ET TEMPORELLE DU SON ; LES DEUX TEMPS DE LA CRÉATION 145

Chapitre 8 Phase de lutherie ; composition d'objets CORDIS-ANIMA 147

8.1 - TÂCHES ÉLÉMENTAIRES 148
8.2 - LA SIMULATION EN PHASE DE LUTHERIE 149
8.3 - STRATÉGIES ITÉRATIVES ET INCRÉMENTALES 150
8.4 - CONCEPTION HIÉRARCHIQUE 151
8.5 - LES NOTIONS D'ÉCHELLE 153
8.6 - LA MACRO-MODULARITÉ DANS LE PROCESSUS DE LUTHERIE 154
8.7 - DÉTERMINATION DES PARAMÈTRES ; EXPLORATION PARAMÉTRIQUE 155
8.8 - RELATIONS ET CHANGEMENT DE BASE DE PARAMÈTRES 156
8.9 - EXPERTISE, ANALYSE ET GÉNÉRATION : VERS UN CONTRÔLE EXPLICITE DES PROPRIÉTÉS 156
8.10 - COMMENTAIRES, LÉGENDES 157
8.11 - DIFFÉRENTES SORTIES DE LA PHASE DE LUTHERIE 158

Chapitre 9 Phase de jeu et autres aspects de la création avec GENESIS 160

9.1 - JOUER AVEC L'INSTRUMENT ; INTERACTION MULTISENSORIELLE VIA LE TGR© 161
9.2 - D'AUTRES FLUX DE DONNÉES 161
9.3 - ENREGISTREMENT ET RELECTURE DE TRACES GESTUELLES 162
9.4 - EDITION DU SIGNAL TRACE : LA BOUCLE D'ÉDITION DE LA TRACE 162
9.5 - D'AUTRES HORIZONS 164

Résumé et Conclusion 165

Conception de GENESIS

Partie IV : Choix généraux pour l'environnement GENESIS

Partie V : Fonctionnalités fondamentales de l'atelier de lutherie GENESIS

Partie VI : Fonctionnalités évoluées du mode lutherie

Partie VII : Simulation et Phénomènes

Conclusions et perspectives

Chapitre 6

UN PARCOURS SIMPLE POUR LA CREATION

Dans ce chapitre, nous effectuons un parcours qui de la conception d'un objet très simple nous permet d'aboutir à la réalisation de séquences musicales – ou, pour le moins sonores. Il s'agit de mettre en perspective *par l'exemple* quelques étapes nécessaires des processus de la création avec CORDIS-ANIMA, de souligner les stratégies attenantes et, parmi elles, de distinguer celles qui sont récurrentes. Pour mener ce parcours, nous supposerons que nous disposons d'un environnement minimal : un calculateur, des périphériques sonores, visuels et gestuels, le langage CORDIS-ANIMA implémenté sous la forme des moteurs de simulation temps réel TELLURIS et temps différé CORDIS-OFF et, enfin, un environnement de programmation.

6.1 - L'objet le plus simple

Exemple	Synthèse
L'objet générateur le plus simple qui puisse être conçu avec CORDIS-ANIMA est l'oscillateur mécanique élémentaire non amorti. Il est constitué de trois modules : un MAS (MASse), un RES (RESsort) et un SOL (point fixe). Pour construire un oscillateur mécanique non amorti il nous faut : • Ajouter ces trois modules à la scène. • Connecter les points L du RES aux points M du MAS et du SOL respectivement. • Affecter des paramètres à chaque module : l'inertie du MAS, l'élasticité du RES – et le cas échéant sa longueur au repos. Nous les choisissons de telle sorte que la fréquence propre de l'oscillateur soit dans le domaine audio. Il faut encore donner de l'énergie à l'objet pour qu'il puisse osciller Pour cela, nous allons simplement affecter des *conditions initiales* aux <MAT> de la scène : soit spécifier que le MAS a une vitesse initiale non nulle, soit le décaler de sa position d'équilibre. A l'issue de ces quelques actions, l'objet est prêt à être simulé. Aucun module d'interaction gestuel n'est présent sur la scène : l'objet est isolé, ses conditions initiales sont seules responsables de son énergie initiale. Nous pouvons donc utiliser le moteur temps différé.	Construction de la structure : Ajout de modules. Pratique des règles de connexion CORDIS-ANIMA. Edition des paramètres des modules de la scène ; Valeurs standards des paramètres. Edition des conditions initiales de chacun des modules <MAT>. Simulation temps différé d'un l'objet isolé.

Exemple	Synthèse
Même lorsqu'elle n'est pas *interactive* la simulation reste *multisensorielle* : elle génère des *phénomènes* dynamiques que nous allons *écouter* bien sûr, mais aussi *visualiser* ou encore *mesurer*.	Observation des *phénomènes.*
Pour *visualiser* l'objet, nous projetons à l'écran, pendant la simulation, les positions du SOL – immobile – et du MAS – mobile et le cas échéant le RES qui les relie. Dans la mesure ou la simulation a lieu en temps-différé, les phénomènes normalement à fréquence audio sont ici ralentis. Ils sont donc observables à l'œil. La visualisation ainsi obtenue peut nous aider à comprendre le fonctionnement de l'objet simulé. Ici, ce fonctionnement est très simple : nous observons un mouvement sinusoïdal.	Evaluation perceptive visuelle : *comprendre* le comportement de l'objet en visualisant ses évolutions au ralenti.
Pour effectuer des mesures, nous captons certains des signaux de force et/ou de position circulant à chaque pas de simulation entre les modules et les traitons ou les analysons de telle sorte que nous puissions en tirer des enseignements – sous forme graphique par exemple. Une telle mesure nous permet de compléter notre connaissance des propriétés de l'objet.	Métrologie : mesurer et analyser les phénomènes générés.
Pour *écouter* les phénomènes générés par notre objet, nous ajoutons un module SOX de sortie sonore et le connectons au MAS de l'oscillateur. Ce module est un <LIA> dégénéré. Il « mesure » la position du <MAT> auquel il est connecté et sait traiter le signal ainsi obtenu comme un signal sonore – l'enregistrer dans un fichier son par exemple. Une fois ce module introduit et paramétré, nous sommes à même d'obtenir notre premier signal sonore. Ses échantillons sont à l'image des évolutions de la position du MAS. Outre qu'elle nous offre quelques nouveaux enseignements sur l'objet (simuler pour comprendre), l'écoute du son constitue notre principal *outil d'évaluation* : c'est elle qui, en définitive, nous permet de *décider de l'intérêt musical des objets*.	Placer les *modules d'écoute.* Ecouter l'objet (évaluation perceptive sonore).
Ainsi, la simulation n'est pas seulement destinée au *jeu* (via le TGR©) ou à la synthèse de sons à vocation exclusivement musicale. Elle s'avère nécessaire au cours de la phase de conception des objets. Elle est au cœur d'un processus consistant à simuler, évaluer puis modifier l'objet en conséquence.	La simulation, une étape essentielle et récurrente.
Cette boucle sera évoquée à plusieurs reprises dans la suite, avant même que ne soit envisagé un quelconque objectif musical. Nous allons d'ailleurs dès maintenant y avoir recours en envisageant des *variations* paramétriques et structurelles toujours autour de notre oscillateur élémentaire.	Simuler : comprendre et évaluer. Boucle modification / évaluation.

Exemple	Synthèse
Substituons, donc, en REF (REssort Frottement) le RES. Nous disposons maintenant d'une nouvelle propriété : l'amortissement. Le nouvel objet est un oscillateur mécanique amorti.	Complexifier la structure. Substituer des modules.
Nous allons maintenant procéder à l'exploration de *l'espace des paramètres* de l'objet.	Exploration de l'espace des paramètres.
Ce faisant, nous mettrons en évidence des *relations* entre cet espace des paramètres et celui des phénomènes générés, par ailleurs générales. Ainsi : • la *raideur* du REF est liée à la *fréquence* du son sinusoïdal généré – plus la raideur est importante, plus la fréquence est élevée. • la *viscosité* du REF contrôle le *coefficient d'amortissement* de l'oscillateur – plus la viscosité est grand plus le temps d'amortissement diminue. • l'*inertie* du MAS apparaît comme un paramètre redondant. Modifiée, elle changera à la fois la fréquence et le temps d'amortissement de l'objet. Avec les structures linéaires, l'inertie n'offre pas un contrôle indépendant du timbre ; elle deviendra cependant de première importance plus loin lorsqu'il s'agira de régler les *interactions* entre différentes structures.	Identification des liens existant entre paramètres et phénomènes. Notamment : *Raideur _ fréquence ; Amortissement _ temps d'amortissement ; Inertie _ impédance.*
Ces *relations* ici découvertes de façon expérimentale peuvent être explicitées mathématiquement (les règles simples dans le cas de notre oscillateur élémentaire peuvent d'ailleurs être généralisées [Incerti.96]). Bénéficiant de cette *expertise*, nous allons *changer notre base paramétrique*, en introduisant dans notre programme les *meta-paramètres fréquence* et *temps d'amortissement*. Pour un ensemble quelconque de valeurs souhaitées pour l'inertie du MAS, la fréquence et le temps d'amortissement, notre programme est maintenant capable de calculer les paramètres CORDIS-ANIMA nécessaires à la simulation.	Relations entre paramètres. Paramètres de *haut niveau* ou meta-paramètres.
Pour mettre un terme à la conception de cet objet élémentaire et pouvoir le réutiliser ultérieurement, ajoutons des *commentaires* à l'ensemble du « programme » réalisé. Comme il se doit, nous séparons les commentaires généraux de ceux qui décrivent les constituants de l'objet et ses meta-paramètres. Enfin, nous décidons de sauvegarder plusieurs *versions* de l'objet (c'est à dire ici du programme) sur disque, pour une éventuelle réutilisation. Ainsi, au fur et à mesure de nos investigations, nous allons progressivement élaborer un *catalogue* d'objets. Ces tâches de commentaires et d'archivage ne seront plus évoquées dans la suite de notre parcours – nous supposerons, cependant, que nous les effectuerons à l'issue de chaque étape.	Commentaires. Versions. Archivage et catalogue.

6.2 - Structures vibrantes

A l'issue du paragraphe précédent, nous disposons d'une *famille d'oscillateurs mécaniques élémentaires*. Bien évidemment, nous ne saurions nous satisfaire d'objets aussi simples et des sons purs que génèrent leurs mouvements et nous allons construire une *structure vibrante* plus complexe.

Exemple	Synthèse
Concevoir une structure CORDIS-ANIMA peut être un exercice difficile. Une stratégie, en tout état de cause, est nécessaire.	
Nous pouvons tout d'abord nous fier à notre imagination pour explorer les possibilités du formalisme – par exemple en ajoutant progressivement de nouveaux modules à l'oscillateur mécanique élémentaire et en *évaluant* l'objet à chaque étape par la simulation. Dans cette approche, la structure physique de l'objet ou son éventuelle interprétation ne nous intéresse guère : les phénomènes qu'il engendre, notamment sonores, retiennent plus notre attention. Il s'agit alors d'une démarche exploratoire et intuitive.	Exploration intuitive *Versus*
Nous préférons adopter ici une démarche plus proche de la *modélisation*. Considérant comme référent réel la corde vibrante, nous allons chercher à en exprimer une *métaphore* dans le langage CORDIS-ANIMA. La corde constitue alors une inspiration pour nos investigations et ce quand bien même notre désir n'est pas d'en proposer une représentation fidèle : ce n'est pas tant la « précision » de la métaphore qui importe (i.e. la distance entre les phénomènes qu'elle génère et ceux du référent) que la démarche que nous mettons en œuvre.	modélisation et recherche de métaphores.
Le réseau CORDIS-ANIMA qui évoque intuitivement la corde est le *chapelet* : N MAS sont équidistantes sur un axe, (N+1) REF les relient deux à deux.	Recherche de métaphore. Pratique de CORDIS-ANIMA
Pour la construire, introduisons les MAS et les RES, connectons les RES aux MAS les plus proches et paramétrons chaque module de telle sorte que le chapelet soit homogène. Pour qu'une *oscillation* autour d'une *position d'équilibre* soit possible, ajoutons des *points fixes* SOL à chaque extrémité du chapelet.	Importance des points fixes.
Nous spécifions ensuite un *point d'écoute* en connectant un module de sortie sonore (<LIA> dégénéré) à l'un des MAS : les mouvements de ce MAS seront transformés en signal sonore. Enfin, conférons une *énergie initiale* au chapelet en affectant une vitesse initiale non nulle à un autre MAS de la corde. Ceci fait, nous pouvons comme précédemment lancer une simulation (à nouveau avec le moteur temps différé, puisque l'objet est encore isolé) et observer les phénomènes – notamment visuels et sonores.	Excitation par rupture de l'équilibre à l'instant initial. Simulation temps-différé.

Exemple	Synthèse
Nous explorerons le *potentiel sonore* de la structure chapelet en nous plaçant dans la boucle désormais classique de modification / simulation / évaluation phénoménologique.	Exploration systématique : Boucle modification / évaluation.
Procédons d'abord à quelques modifications d'ordre structurel en déplaçant le point d'écoute. Il apparaît que le choix du MAS « écouté » n'est pas indifférent : en écoutant le chapelet au centre, nous favorisons dans le son les *modes* pairs du chapelet (ceux qui ont un ventre au centre) et notamment le mode fondamental. En écoutant au bord, nous favorisons les modes impairs. La position d'écoute nous offre ainsi un contrôle du *timbre* des sons générés.	Légères variations de la structure. Déplacement de connexions.
La même propriété apparaît si nous modifions le lieu de l'excitation en rompant l'équilibre au niveau d'un autre MAS. Notons que les conséquences d'un tel déplacement des points d'écoute et / ou d'excitation sur les phénomènes est à l'image de ce qu'autorisent les instruments réels[1]1 : la métaphore de la corde nous permet de tirer parti efficacement de notre expérience des cordes vibrantes réelles, *de projeter cette expérience de façon intuitive sur l'objet CORDIS-ANIMA*.	Projection de l'expérience du musicien dans l'univers CORDIS-ANIMA.
Ainsi, pour faire varier les propriétés de notre corde, il n'est pas nécessaire d'agir sur les paramètres. Nous pouvons plus simplement modifier sa longueur en ajoutant ou supprimant des modules, à la manière de ce que fait le violoniste lorsqu'il déplace ses doigts sur la touche. En doublant la longueur du chapelet, par exemple, nous diviserons par deux la fréquence fondamentale des sons qu'elle génère.	
Une longueur étant choisie, nous pouvons commencer l'exploration des paramètres du modèle.	Exploration paramétrique systématique.
Nous conserverons pour commencer l'homogénéité du chapelet ; il nous faut donc faire varier *ensemble* les paramètres d'inertie des MAS d'une part, ceux de raideur et la viscosité des REF d'autre part. Ici apparaît pour la première fois la notion de *groupes morphologiques de modules* : nous considèrerons les modules en *groupes* de paramètres *homogènes* et dès lors nous n'utiliserons que les trois paramètres (inertie, raideur, viscosité) du groupe. C'est là, cependant, une facilité temporaire pour l'édition : nous voulons également continuer à considérer le chapelet comme une collection de modules et accéder individuellement à chacun d'entre eux (ainsi qu'à ses paramètres) – pour, par exemple, déplacer le point d'écoute ou le point d'excitation ou encore pour rompre l'homogénéité si telle est notre volonté	Modularité morphologique. Edition des paramètres homogènes. Conserver un accès *individuel* à chacun des modules.

[1] Pensons, par exemple, au guitariste qui fait varier constamment la distance de sa main droite au chevalet.

Exemple	Synthèse
Comme lors de l'étude de l'oscillateur, l'exploration des trois paramètres du chapelet homogène va mettre en évidence l'existence d'un lien prédictible entre l'espace des paramètres et celui des phénomènes. Plus précisément, les relations *raideur* ↔ *fréquence fondamentale* et *viscosité* ↔ *temps d'amortissement* apparaîtront.	Relation entre paramètres physiques et phénomènes, ou « paramètres perceptifs ».
Nous détaillerons plus loin les calculs qui, appliqués à une structure CORDIS-ANIMA, permettent de caractériser ses modes de vibration [Incerti.96], [Djoharian.97]. Pour l'heure, nous supposerons que nous disposons des outils nécessaires pour réaliser *l'analyse modale* du chapelet – ou, du moins, de ses modes transversaux.	Analyse modale. Connaissance des modes : fréquence, temps d'amortissement, déformée modale.
Avec l'analyse modale nous pouvons par exemple déterminer l'importance relative de chaque mode sur chacun des MAS du chapelet – apportant ainsi une justification théorique des résultats que nous avons obtenus précédemment quant à l'influence de la position d'excitation et d'écoute – ou encore estimer le « degré d'harmonicité » du chapelet. Elle nous permet en outre *d'accorder* le chapelet, c'est à dire de calculer ses paramètres de raideur (resp. de viscosité) de telle sorte que son mode fondamental oscille à une fréquence (resp. avec un temps d'amortissement) donné – les autres modes étant modifiés en conséquence[1].	Accordage.
D'une certaine manière, l'accordage génère des paramètres (en fait, il s'agit dans ce cas plus précisément d'une modification) de telle sorte que la fréquence fondamentale du son soit à une fréquence donnée. Dans un même ordre d'idées, évoquons ici la possibilité *théorique* de générer de façon automatisée une structure vibrante avec ses paramètres, de telle sorte que certaines de ses propriétés satisfassent un ensemble de valeurs souhaitées. Avec un tel procédé de *génération de structure paramétrée* il devient par exemple possible de déterminer un objet CORDIS-ANIMA qui permette la re-synthèse d'un son.	Génération de structures paramétrées.
Quoi qu'il en soit, dans le cas assez simple du chapelet homogène, les relations entre paramètres d'une part, fréquence fondamentale et temps d'amortissement d'autre part, peuvent être déterminées algébriquement [Incerti.96]. Explicitons, donc, ces relations dans notre programme en introduisant les *meta-paramètres fréquence* et *temps d'amortissement* ainsi qu'un algorithme calculant les paramètres CORDIS-ANIMA à partir des valeurs que nous aurons saisies.	Changement de base de paramètres : relation et meta-paramètres.

[1] L'accordage peut, en fait, être effectué sur n'importe quel mode. Notons qu'il s'agit ici d'une modification paramétrique qui ne doit pas être confondue avec l'accordage que pratique le musicien lorsqu'il modifie la tension d'une corde – sans modifier sa matière.

Exemple	Synthèse
Incidemment, nous pourrions également définir un *meta-paramètre* par le rapport raideur/amortissement. Fixé à une valeur donné, il nous assurerait dans une certaine mesure que le chapelet évoque une « matière » donnée – tel que le boyau, métal… - lorsqu'on change sa raideur[1]. Plus généralement, rien n'impose que les meta-paramètres aient une quelconque signification « physique » : il peut être intéressant de pratiquer très largement le changement de base paramétrique à l'aide de relations et de meta-paramètres. L'exploration du chapelet pourrait se poursuivre. Nous la clôturons ici, non sans avoir commenté le programme et archivé les objets (paramétrés) que nous voudrons conserver. Rappelons, enfin, que l'exemple du chapelet n'est pas limitatif. Nous pourrions obtenir d'autres métaphores – des instruments à vent, des peaux, des barres ou d'autres structures vibrantes réelles – ou, dans une démarche plus « exploratoire et intuitive », laisser libre cours à notre imagination pour construire avec CORDIS-ANIMA des structures vibrantes inouïes.	Commentaires, archivage.

6.3 - Com-position de structures vibrantes. Chevalet, environnement local, environnement global.

Exemple	Synthèse
Nous disposons maintenant de plusieurs chapelets. En les insérant dans un modèle vierge, en réitérant le processus du paragraphe précédent ou encore en insérant un objet depuis un catalogue, nous pouvons poser sur la scène *plusieurs* structures vibrantes. Nous voulons, cependant, les *com-poser* – littéralement les *poser ensemble* – c'est à dire les intégrer au sein d'un objet plus complet.	Catalogue. Insertion depuis le catalogue.
Pour composer les structures, nous allons nous inspirer des scènes instrumentales traditionnelles. Dans la famille des cordes, par exemple, différentes structures vibrantes sont posées sur un même *chevalet* qui les met en interaction. Le chevalet transmet leurs vibrations à un résonateur ou *environnement local* qui permet la transmission de la vibration à l'air (adaptation d'impédance). Enfin, les différents instruments de l'orchestre sont plongés dans un *environnement global* (la salle de concert par exemple) qui a son tour impose sa signature au son qui parvient finalement aux oreilles des auditeurs.	Analyse des scènes réelles. Structures vibrantes, Chevalet, Environnement local, Environnement global.

[1] Le rapport Z/K permet une première évocation de la matière. Voir [Djoharian.97].

Exemple	**Synthèse**
Pour continuer notre parcours, nous allons chercher une métaphore pour chacun des constituants ainsi mis en évidence : chevalet, environnement local, environnement global. Ce faisant, nous allons introduire une *hiérarchie* au sein des objets CORDIS-ANIMA.	Recherche de métaphores Construction hiérarchique en aval des structures vibrantes.
La « métaphore » la plus simple qu'on puisse trouver pour le chevalet est un oscillateur élémentaire (composé d'un SOL, d'un MAS et d'un REF) dont la viscosité est suffisamment élevée pour que l'amortissement soit critique. Excité, un tel oscillateur revient à sa position d'équilibre sans oscillation. Lorsqu'il est soumis à une force sinusoïdale de fréquence variable, sa réponse en amplitude est presque plate : il ne favorise ou ne défavorise aucune fréquence. Incidemment, l'obtention d'un amortissement critique est assurée par une *relation* entre les paramètres d'inertie, de raideur et de viscosité ; nous pourrions, à nouveau, expliciter cette relation de telle sorte que par exemple la viscosité de notre chevalet soit calculée à partir des autres paramètres.	Chevalet. Amortissement critique. Relation.
Insérons, maintenant, deux chapelets sélectionnés par exemple parmi ceux que nous avons précédemment conçus, supprimons les points fixes SOL à l'une de leurs extrémités, puis connectons les points L devenus libres des REF à notre chevalet. Pour écouter cet objet *composé*, plaçons le module de sortie sur le chevalet.	Insertion d'objet. Com-position. Modifications structurelles.
Observons un instant notre nouvel objet. Nous avons posé ensemble plusieurs structures et ainsi réalisé une sommation au niveau du chevalet, par le biais d'une construction physique, de certains des phénomènes dont elles sont le siège. Cependant, nous avons aussi mis ces structures en interaction, l'une pouvant agir sur l'autre via le chevalet. Désormais, les chapelets s'influenceront mutuellement par sympathie. Cette interaction est à l'image d'une interaction possible entre objets réels. Comparée à un simple montage sonore, elle garantit une *plausibilité* accrue aux sons générés.	La « com-position » implique interaction et rétroaction. Elle garantit la plausibilité des phénomènes générés.
Il nous faut cependant maîtriser les effets de l'interaction par un choix précis de paramètres tant pour les chapelets que pour le chevalet.	Maîtrise des interactions entre structures.
Procédons, d'abord, à la *mise à l'échelle* des chapelets. Pour modifier l'échelle *d'impédance* d'une structure – ici l'un des chapelets – nous pouvons multiplier tous ses paramètres physiques (raideur, inertie, viscosité) par un *même coefficient* : ses propriétés intrinsèques restent inchangées, mais son aptitude à échanger de l'énergie avec l'extérieur est, elle, modifiée. Ici, il faudra que les deux chapelets aient des inerties comparables, c'est à dire par exemple que leurs MAS respectifs aient des inerties égales. Incidemment, il pourrait être intéressant de définir un meta-paramètre par le quotient entre les inerties des MAS de chacun des chapelets, puis d'explorer les conséquences phénoménologiques et sonores de variations de ce quotient.	Maîtrise des échelles d'impédance. Modifier les paramètres par des fonctions. Ici : *multiplier par un coefficient.*

Exemple	**Synthèse**
Il nous faut ensuite mettre le chevalet à l'échelle des structures vibrantes. En explorant les paramètres du chevalet, nous verrions que nous pouvons maîtriser l'interaction entre les deux chapelets, c'est à dire l'importance du phénomène de vibration par sympathie. Un facteur 100 entre l'inertie des MAS des chapelets et celle du chevalet garantit à la fois que le chevalet soit équivalent à un point fixe pour les chapelets et qu'une interaction sympathique faible mais perceptible existe. Nous le choisirons donc. Notons qu'en conséquence, pour une amplitude donnée des vibrations des chapelets nous recueillerons sur le chevalet une amplitude bien moindre – ce qui d'ailleurs est à nouveau est à l'image des phénomènes observables sur les instruments réels.	Exploration. Boucle modification / simulation. Echelles relatives : importance des *rapports* entre paramètres des différentes structures.
Nous laissons le lecteur élaborer puis connecter au chevalet l'environnement local qu'il lui plaira, étant entendu que sa construction n'introduira pas de nouvelles dimensions à notre propos – les tâches en cause restant du même type. L'objet ainsi obtenu est doté de plusieurs structures vibrantes en interaction et d'un environnement local. Il constitue une métaphore d'un instrument complet.	Environnement local. Instrument.
Que se passe-t-il si nous insérons dans la scène un second objet du même niveau que le précédent dans la hiérarchie des objets ? Nous avons alors deux instruments indépendants. Pour les composer, nous allons construire un environnement global très élémentaire.	Environnement global.
Dans le monde réel, l'action d'un instrument sur un autre à travers le milieu aérien est faible. La *fonction* de l'environnement global est d'effectuer une sommation des mouvements des instruments présents sur la scène, mais plus nécessairement de les mettre en interaction.	
Nous allons ici représenter l'environnement global au moyen d'un nouvel oscillateur élémentaire à son amortissement critique, doté d'une inertie très faible relativement à celle des instruments. Ajoutons-le à la scène puis connectons-le aux chevalets – c'est-à-dire aux environnements locaux – des divers instruments à l'aide de liaisons *faibles* et exclusivement visqueuses (i.e. : des modules FRO). Enfin, déplaçons le module d'écoute sur le MAS du nouvel oscillateur pour « écouter » ses mouvements.	Maîtrise des interactions entre objets.
Un tel environnement global permet, en quelque sorte, de sommer les *vitesses* des MAS auquel il est connecté puisque la force exercée par les FRO est proportionnelle à la somme des vélocités de chaque chevalet. Réciproquement par contre, il influence très peu les structures vibrantes qu'il « compose », du fait de l'inertie et de la viscosité que nous avons choisies.	Impédance et échelles d'inerties.

Exemple	**Synthèse**
Observons l'objet obtenu. C'est un ensemble de modules, mais c'est aussi un ensemble de *sous-objets* cohérents, chacun ayant une *fonction* et une *position* dans la hiérarchie de l'objet complet. Cet exemple fait ainsi apparaître la notion de *modularité fonctionnelle* (ou macro-modularité), notion essentielle pour le processus de création avec CORDIS-ANIMA.	Modularité typologique ou fonctionnelle.
Tôt ou tard, nous voudrons rendre plus explicites ces sous-objets, en les considérant non plus comme des collections de modules élémentaires – auxquels il resterait possible d'accéder individuellement – mais précisément comme des *macro-modules*. En d'autres termes, nous serons amenés à cacher le niveau des modules CORDIS-ANIMA et de leurs paramètres, pour considérer tel ou tel ensemble de modules comme un module de niveau supérieur, doté de paramètres spécifiques (par exemple, dans le cas d'un chapelet homogène, les trois paramètres inertie, raideur et viscosité). Par ailleurs, il nous faudra aussi exprimer des *relations* et définir des *meta-paramètres* non plus au sein d'un même objet mais entre les paramètres que nous aurons retenus pour différents macro-modules.	Macro-modules. Relations entre meta-paramètres.

6.4 - Excitateur, modificateur, déclencheur

Exemple	**Synthèse**
Dans les exemples précédents, les objets ont été excités de la façon la plus simple possible : ce sont les *conditions initiales* d'un MAS qui, rompant l'équilibre de l'objet étaient responsables d'une *énergie initiale* non nulle et de leurs mouvements.	Observation d'objets réels.
Ce procédé d'excitation n'est guère satisfaisant d'un point de vue sonore : les sons générés ne sont pas naturels. De fait, il n'est pas *à l'image* de ce que nous ferions spontanément face à un objet réel. Les structures vibrantes réelles sont naturellement au repos. C'est au contact d'un autre corps – l'excitateur - qu'elles acquièrent de l'énergie, au cours par exemple d'une interaction de frottement, percussion ou pincement. Nous allons maintenant construire quelques métaphores d'un tel *corps à corps.*	Recherche de métaphores des *corps à corps.*

Exemple	Synthèse
Nous commencerons en réalisant une *percussion*. Cela suppose que nous puissions exprimer une interaction conditionnée à la position relative de deux « corps » : le percuteur et le percuté n'interagissent que lorsqu'ils sont suffisamment proches. Le module BUT (BUTée viscoélastique) s'impose ici naturellement. Pour *percuter* la structure vibrante, nous introduisons un nouveau MAS, le *percuteur,* puis un BUT qui sera connecté d'une part au MAS, d'autre part à la structure vibrante. En lançant le MAS *vers* la structure vibrante, nous observerons : • Une phase d'incidence, pendant laquelle le percuteur se rapproche de la structure vibrante sans qu'il n'y ait interaction effective – la force développée par le BUT est nulle. La structure est au repos. • Une phase d'interaction, qui débute lorsque que le percuteur et le MAS sont suffisamment proches. Cette interaction peut être plus ou moins longue et plus ou moins complexe. • Puis à nouveau une phase sans interaction, dès lors que le percuteur est rejeté. Le percuteur redevient *libre* ; il conserve sa lancée et repart vers les limites de l'espace de simulation. Avec le percuteur, de nouveaux paramètres apparaissent : inertie du MAS, raideur, viscosité, taille de la « sphère d'influence » ou « seuil » du BUT. Ces paramètres nous permettent de travailler la *qualité* de la percussion – durée, caractère, possibilité ou non d'oscillation pendant la percussion– et ainsi, pour ce qui est du phénomène sonore, sculpter la qualité de *l'attaque.*	Percuteur élémentaire. Pratique de CORDIS-ANIMA.
Lors de l'exploration des paramètres (mais aussi de façon théorique, voir [Fourcade.01]), l'inertie M du percuteur et le rapport R entre inertie du percuteur et raideur du BUT constituent une base de paramètres intéressante : à M on peut associer la *brillance* de l'attaque et à R sa *durée*, c'est à dire la durée de la phase d'interaction effective entre percuteur et percuté. Nous voudrons alors disposer de M et R en tant que paramètres, c'est à dire définir un meta-paramètre R.	Exploration paramétrique. Maîtrise de l'interaction entre structures. Relations et meta-paramètres.
Les conditions initiales du percuteur offrent quant à elles : • Un contrôle de l'énergie cinétique du percuteur (vitesse initiale) donc de l'énergie transmise à la structure vibrante lors du choc et de l'amplitude de l'oscillation de la structure vibrante. • Un contrôle de l'instant auquel se produira la percussion : le percuteur entrera en interaction d'autant plus tard qu'il est placé loin de la structure au début de la simulation. Pour une vitesse (ou une énergie) donnée, la position initiale du percuteur détermine la date d'excitation, qui peut être calculée.	Utiliser les conditions initiales pour spécifier la date et l'intensité d'un *événement sonore* généré par un percuteur.

Exemple	Synthèse
Le principe de la percussion, ici donné en exemple, peut être généralisé. Nous appellerons alors *déclencheur* un ensemble MAS / LIA qui lancé à une certaine distance de la structure, viendra modifier son comportement à une date donnée de son histoire.	Notion de déclencheurs.
Reprenons notre précédent objet. Il est possible de lui connecter un nombre quelconque de déclencheurs, chacun venant tour à tour percuter une structure vibrante, l'amortir (BUT uniquement visqueux), l'exciter à la manière d'un plectre (liaison non linéaire particulière) ou plus généralement modifier son état énergétique à certaines dates de son histoire. En réglant les conditions initiales et les paramètres des déclencheurs, nous maîtrisons les évènements sonores générés tant du point de vue temporel qu'en termes de qualité sonore. Avec nos structures vibrantes composées et une *partition de déclencheurs*, nous sommes donc en mesure de « composer de la musique » - au sens traditionnel d'agencement d'évènements sonores préalablement à leur genèse.	Jouer avec des déclencheurs. Partition de déclencheurs.

6.5 - Instrumentiste, chef, déclencheurs

Exemple	Synthèse
En matière de génération d'événements sonores par des moyens ou « métaphores » physiques, une approche plus complète est possible[1]. Considérons pour commencer un objet CORDIS-ANIMA (un chapelet par exemple) et réglons ses paramètres de telle sorte que ses oscillations aient lieu à basse fréquence – typiquement de l'ordre du Hertz. Une telle structure ne peut plus générer de son audible : le signal résultant de la capture du mouvement d'un de ses <MAT> est dans le domaine sub-audio. Nous dirons qu'elle génère des *évènements*, dans la mesure ou son comportement en cours de simulation est perceptible comme tel, par le canal visuel par exemple.	Echelle fréquentielle.

[1] La possibilité d'une *composition d'objets physiques* telle que celle à laquelle nous allons procéder dans ce paragraphe est notamment démontrée par les récents travaux de Claude Cadoz [Cadoz.02a], [Cadoz.02b].

Exemple	Synthèse
Connectons alors cet objet basse-fréquence à une structure vibrante à l'aide d'un module BUT viscoélastique ou, le cas échéant, à l'aide d'un excitateur intermédiaire. A chaque fois que le <MAT> connecté de l'objet basse fréquence entre dans la sphère d'influence du <MAT> connecté de la structure vibrante une percussion se produit.	Modularité typologique.
Ainsi, les phénomènes générés par l'objet sont transformés en phénomènes successifs de nature acoustique. Si nous « écoutons » la structure vibrante, nous pourrons entendre les évènements sonores résultant de l'interaction des deux structures, en plus du timbre propre à la structure vibrante.	
Nous dirons en conséquence que l'objet basse-fréquence est une métaphore d'un *instrumentiste*. Comme précédemment, nous pouvons modifier la structure de l'instrumentiste et ses paramètres – et par exemple élire des groupes morphologiques, définir des relations ou des meta-paramètres, etc... Ce faisant cependant, nous ne maîtriserons plus le timbre des sons générés, mais au contraire deux nouvelles dimensions phénoménologiques, à savoir :	Instrumentiste.
• L'organisation temporelle des évènements générés – de par les propriétés intrinsèques de l'instrumentiste.	L'instrumentiste « est » la partition.
• La qualité de l'interaction entre l'instrumentiste et son instrument et notamment la possible rétroaction de ce dernier sur l'instrumentiste. Au contact de la structure, l'instrumentiste peut perdre – ou non – son énergie. De façon générale, en augmentant l'inertie de l'instrumentiste nous diminuons son aptitude à être influencé par la structure vibrante.	Maîtrise de l'interaction entre échelles.
La démarche précédente peut être réitérée : l'instrumentiste, à son tour, est alors « soumis » à un nouvel objet, de masse plus importante encore et de fréquence plus basse, qui, par ses mouvements, va contrôler les évènements générés avec une plus grande constante de temps. Nous dirons alors que nous disposons d'une métaphore d'un *chef*.	Modularité typologique. Chef.
En ajoutant encore à cette construction hiérarchique des déclencheurs, il est possible de déclencher et/ou modifier les chefs, les instrumentistes ou les structures vibrantes à des dates précises. La construction hiérarchique en amont des structures vibrantes peut être arbitrairement complexe. En la travaillant, nous pouvons générer et maîtriser à la fois les timbres des sons générés, les évènements ou des séquences sonores plus complexes. De ces séquences sonores aux *séquences musicales* la distance n'est plus guère importante.	Construction hiérarchique en amont de la structure vibrante.

6.6 - Fin de la phase de conception

A l'issue de la phase de conception, qui peut incidemment s'arrêter à un niveau de complexité quelconque, nous avons obtenu des *sons* et des *familles d'instruments*. Lorsqu'un objet nous semble suffisamment satisfaisant, nous voudrons commencer à en jouer. Nous quittons alors la phase de conception pour entrer dans une phase de jeu avec l'objet désormais stabilisé. Nous nous intéressons alors aux flux dynamiques qui permettront, parmi tous les sons que l'objet peut générer, d'en choisir un en particulier.

Avant d'entrer dans cette phase de jeu, apportons une première conclusion aux paragraphes précédents.

Ils ont mis en évidence la possibilité d'un processus de création *exclusivement* basé sur la construction et la simulation d'objets physiques CORDIS-ANIMA et permettant d'aboutir à une œuvre sur support. Avec ce processus « complet », nous avons substitué à la composition traditionnelle la notion de *composition d'objets physiques.*

L'équivalent de la partition est alors l'objet physique complet (et complexe), contenant plusieurs sous-objets de niveaux hiérarchiques divers, munis de leurs paramètres et conditions initiales. Remarquons, enfin, qu'à aucun moment nous n'avons eu recours jusqu'ici à une interaction temps réel avec les objets : la simulation temps différée est suffisante pour la mise en œuvre du processus de composition d'objets. Le jeu procède alors de la maîtrise des conditions initiales (notamment pour les déclencheurs), données temporelles élémentaires qui permettent de contrôler la date des événements et leur intensité.

6.7 - Jeu et interaction multisensorielle

Exemple	Synthèse
Envisageons, maintenant, la situation d'interaction multisensorielle via le TGR©. Pour qu'elle soit possible, il est bien sûr nécessaire que l'objet soit suffisamment simple – ce que nous supposons. Ajoutons, donc un ou plusieurs *modules d'interaction* à l'objet, points d'accès à l'objet et image du ou des touches du transducteur[1]. La position de ces modules d'interaction dans la hiérarchie des structures peut être choisie arbitrairement : nous pouvons décider de les connecter par exemple à un excitateur, un instrumentiste ou un chef. Ceci fait, nous pouvons toucher, voir et entendre l'objet simulé en temps réel. On comprendra que pour « jouer de l'objet » un apprentissage parfois difficile soit nécessaire, de la même manière qu'on apprend à jouer du violon. On peut par ailleurs imaginer qu'une partition – au sens traditionnel cette fois ci – puisse être écrite pour chaque instrument simulé et ses périphériques. Nous n'insisterons cependant pas plus sur la situation d'interaction temps réel : si son intérêt est évident, elle n'est guère différente de la situation instrumentale traditionnelle et n'appelle pas plus de précision ici.	Simulation temps réel. Connexion au TGR©. Apprentissage de l'instrument.

6.8 - L'objet " trace " et l'incrémentation du jeu

Exemple	Synthèse
L'innovation majeure tient en fait à la possibilité *d'objectiver* les gestes effectués en temps réel en enregistrant les *traces gestuelles*. La suite de notre parcours mettra en évidence les possibilités offertes par une telle situation. Lors d'une simulation, enregistrons sur disque une première trace gestuelle. En appliquant cette trace enregistrée lors d'une seconde simulation du même objet, nous obtenons les mêmes effets. Nous pouvons donc nous consacrer au jeu sur une autre partie de l'objet. Ce faisant, chose intéressante, nous *incrémentons* le jeu. Nous voudrons toutefois l'étendre en allant plus avant dans la manipulation des traces.	Enregistrement de trace. Relecture. Incrémentation du jeu.

[1] Voir la partie I chap. 1. Précisons en outre que le recours à d'autres transducteurs et standards de communication temps-réel n'est pas exclu.

6.9 - Edition de la trace

Exemple	Synthèse
Une trace est un *signal* : lors de son édition, le temps est une dimension *explicite* suivant laquelle se succèdent des échantillons. A chaque segment correspond à la fois un *geste réalisé* et un *effet obtenu* lors de son application à un objet (disons : un potentiel sonore). Ainsi, nous pouvons envisager deux types de représentation symbolique des traces : en référence à la *perception* ou en référence à *l'action* [Ramstein.91]. Parallèlement, deux approches de l'édition de la trace sont possibles.	Edition des traces en référence à la cause / en référence à la perception.
Pour commencer, *segmentons* la trace afin de n'en conserver que les parties signifiantes et de supprimer les autres – soit qu'elles sont sans effet (par exemple l'excitateur n'est pas en contact avec la structure vibrante), soit qu'elles ne nous satisfont pas. Cette étape réalisée, nous disposons de plusieurs *segments* de traces dans un catalogue. Nous pouvons envisager de reconstituer une trace par concaténation. Il s'agit alors d'une opération de *composition* de traces, qui n'est pas sans parallèle avec la composition au sens traditionnel[1].	Segmentation. Catalogue de traces, composition de traces.
Les opérations de segmentation et de concaténation que nous avons appliquées à la trace restent élémentaires. Elles relèvent du *montage* mais pas de la *transformation.* Nous allons maintenant appliquer à un segment de trace des opérations plus complexes.	
Modifions, par exemple, son amplitude, étirons-la dans le temps ou encore renversons-la. Toutes les transformations connues en traitement du signal sont ici envisageables. Un tel processus n'est pas sans parallèle avec la *démarche concrète* : nous le conduirons en laissant notre perception juger de la validité de chacune des traces obtenues par transformation – gardant toutefois à l'esprit que la trace doit être *appliquée* à un objet physique lors d'une simulation pour devenir perceptible par les canaux sensoriels visuels et auditifs[2].	Transformations. Travail sur le signal.

[1] La note de musique, en effet, est symbole à la fois de certaines caractéristiques voulues pour le son et des gestes qui doivent être réalisés pour l'obtenir. En composant les traces, nous assemblons à la fois les gestes et les effets comme nous composerions des notes – en particulier préalablement à leur « exécution ».

[2] On peut envisager de « percevoir » un signal de trace directement par le canal gestuel via le TGR©. Considérant nos objectifs essentiellement musicaux, on comprendra que la trace est intéressante de par ses effets sur le son – qui ne seront perceptibles que lorsqu'elle est appliquée à un objet CORDIS-ANIMA.

Exemple	Synthèse
Nous allons pour finir mettre en œuvre une autre approche plus originale, cette fois-ci en référence à *l'action* ou à la *cause*. Pour ce faire, nous allons concevoir un objet CORDIS-ANIMA dont les propriétés physiques « auraient pu » générer la trace enregistrée, c'est-à-dire un objet qui permette la « re-synthèse » de la trace ou encore la « simulation » de l'instrumentiste. Il est certain que l'obtention d'une telle représentation causale n'est pas aisée. Nous pouvons, cependant, nous aider du principe de la *génération automatisée* déjà évoqué. A supposer que nous ayons pu obtenir une représentation causale de la trace, nous disposons d'un nouvel objet répondant au formalisme CORDIS-ANIMA. Nous pouvons retourner à une nouvelle phase de lutherie pour à son tour le modifier. Ce faisant, nous réalisons bien une « édition et une transformation de la trace ». Par contre, puisque nous travaillons au niveau des propriétés physiques d'un « objet, » nous pouvons espérer obtenir des variations autour de la trace originelle qui soient pertinentes pour la perception. Mais ce n'est pas tout. L'objet physique qui représente la trace conserve sa partie signifiante : il détermine par exemple une percussion, plusieurs percussions successives et de force variable, un rythme, etc. Ces évènements, cependant, sont eux *générés*. Il se produit à nouveau lors de la simulation une *interaction* entre instrument et instrumentiste – quand bien même ce dernier est « virtuel ». Cette interaction est à l'image de celle qui existait lors de l'enregistrement de la trace via le TGR©. Elle tend à garantir la cohérence des phénomènes et ainsi une certaine « richesse vivante » [Cadoz.88b].	Représentation causale de la trace. Génération « d'instrumentistes ». Modification d'instrumentistes CORDIS-ANIMA ; « édition causale » de la trace. Simulation de l'interaction instrumentiste / instrument.

Le parcours que nous avons effectué pourrait être complété. Il n'évoque par exemple ni des tâches relevant de l'organisation des divers catalogues (de trace, de structures vibrante, d'instrumentistes, etc.), ni le principe de la modification structurelle et paramétrique dynamique (partie I chapitre 1) pas plus que la possibilité d'un travail à un plus haut niveau d'abstraction, dans lequel des objets, des traces, des segments de trace seraient tous considérés comme des « objets musicaux » et assemblés, « composés » de façon plus abstraite.

Ce chapitre à toutefois d'ores et déjà fait apparaître les démarches et stratégies fondamentales que GENESIS devra soutenir. De ce parcours à vocation démonstrative, il nous faut maintenant passer à des considérations plus générales. Les trois chapitres à venir en proposent donc une *synthèse*. Ils constituent une *analyse globale des tâches* impliquées dans l'utilisation de GENESIS.

Chapitre 7

CONSIDERATIONS GENERALES SUR LA CREATION AVEC CORDIS-ANIMA

Dans ce chapitre :

- Nous précisons ce que peuvent être les objectifs généraux de l'utilisateur lorsqu'il décide d'avoir recours à CORDIS-ANIMA. Nous rappelons en outre qu'il est possible de placer l'outil GENESIS au cœur de la création. La suite se place dans cet objectif englobant.
- Nous montrons que créer avec CORDIS-ANIMA ce n'est pas forcément avoir recours aux simulations temps réel. En conséquence, notre analyse doit être fondée sur le paradigme de la simulation d'objets physiques et non exclusivement sur la situation interactive et multisensorielle.
- Nous montrons que le travail de l'utilisateur confronté à CORDIS-ANIMA s'organisent en deux phases : la phase de lutherie et la phase de jeu. Les deux chapitres suivants sont consacrés successivement aux tâches impliquées dans chacune de ces deux phases.

7.1 - GENESIS, un environnement complet

Du parcours que nous avons effectué, nous pouvons déduire deux objectifs possibles et non exclusifs pour un utilisateur qui désire utiliser GENESIS :

- Il peut vouloir essentiellement jouer avec un objet CORDIS-ANIMA spécifique comme il jouerait d'un instrument traditionnel, par exemple en situation de concert. Il n'est dans ces conditions guère intéressé par la conception de l'instrument, mais plus par la qualité de la relation instrumentale que permet l'ensemble de l'outil.
- Il peut souhaiter mettre en œuvre plus largement l'outil et notamment vouloir créer ses propres objets CORDIS-ANIMA.

Dans cette seconde approche, CORDIS-ANIMA et GENESIS s'insèrent *à priori* parmi d'autres outils du studio audionumérique : synthétiseurs, séquenceurs, tables de mixage, logiciels de montage et de traitement, logiciels de CAO par exemple. Les données échangées avec ces autres machines et logiciels sont alors de première importance en ce qu'elles caractérisent efficacement la place qu'occupe GENESIS dans le processus de création complet de l'utilisateur.

CORDIS-ANIMA peut d'abord être envisagé comme une technique de synthèse. Dans cette approche, les sons générés sont élémentaires. Ils sont exportés puis assemblés dans d'autres environnements (typiquement des logiciels de montage et de traitement). La *composition* est opérée d'une façon traditionnelle, à la manière de la démarche concrète. *Jusqu'à récemment, c'est ce processus qui a été mis en œuvre presque exclusivement* sur les versions successives de GENESIS, du fait notamment de leurs limitations.

Le son numérique n'est pas la seule catégorie de données couramment manipulée dans les environnements audio-numériques : le signal MIDI, les données spectrales, les paramètres des machines de synthèse et traitement sont par exemple également d'usage fréquent. Pour compléter le potentiel de GENESIS en tant que logiciel de studio, il nous appartiendra de distinguer les différents standards avec lesquels il doit être susceptible de communiquer avec son environnement, pendant la simulation mais aussi avant ou après elle, c'est-à-dire *d'ouvrir* le logiciel.

Le chapitre précédent a cependant montré qu'avec la simulation interactive et multisensorielle d'objets physiques il est possible de réaliser dans une même session les étapes de génération du matériau sonore et celles qui relèvent de son organisation. GENESIS peut ainsi se situer au cœur de *l'ensemble* du processus de création et devenir ainsi l'outil central du studio.

Cette approche englobe les précédentes, rend compte au mieux de la diversité des possibilités offertes par la simulation d'objets physiques et garantit qu'un maximum d'innovation soit introduit au sein des processus de la création. Elle est donc particulièrement indiquée pour être le support de l'analyse des tâches qui nous préoccupe.

En conséquence, sans oublier la nécessité de permettre l'échange de différents types de données entre GENESIS et d'autres environnements, les tâches, démarches et stratégies que nous considérons dans la suite sont celles d'un utilisateur qui voudrait :

- Aboutir à la création d'une œuvre sur support finalisée ;
- Envisager GENESIS comme l'outil central du studio audionumérique, support de l'ensemble de son processus de création.

7.2 - Situation temps-réel et temps-différé ; la simulation d'objets physiques

Dans la mesure où l'interactivité et la multisensorialité figurent parmi les concepts fondateurs de nos travaux, le recours au moteur temps-réel semble en première approche une nécessité.

Le parcours du chapitre précédent a cependant montré la possibilité d'un processus de création fondé exclusivement sur la *composition d'objets CORDIS-ANIMA*. Un tel processus fait appel à une hiérarchie étendue des objets en amont de la structure vibrante et peut être conduit avec l'utilisation du seul moteur temps différé. En conséquence, la situation d'interaction multisensorielle basée sur une simulation temps réel n'est en rien systématiquement *nécessaire* à la création avec CORDIS-ANIMA.

Par ailleurs, il existe une limite à la complexité des objets pouvant être simulés en temps réel[1]. Il n'est pourtant pas envisageable de limiter la complexité des modèles que l'utilisateur peut créer – la composition d'objets que nous avons retenue comme l'une des démarches possibles de l'utilisateur nécessite, par exemple, un grand nombre de modules. Il apparaît donc que la situation d'interaction multisensorielle basée sur une simulation temps réel n'est pas systématiquement *possible*.

[1] Typiquement, sur la machine Silicon-Graphics quadri-processeur utilisée au laboratoire, un maximum d'une centaine de MAT et LIA peuvent aujourd'hui être simulés simultanément. Si on peut espérer qu'elle évolue assez rapidement de manière favorable, des années seront sans aucun doute nécessaires avant que le temps réel n'impose plus de limite drastique à la complexité des objets simulables.

Enfin, si les recherches actuelles en matière d'Informatique Musicale tendent de façon presque systématique à généraliser le temps-réel, il convient de rappeler qu'il est loin de résoudre tous les problèmes. Comme l'explique Risset :

> « Le temps différé peut être envisagé comme le contraire d'une limitation : la genèse et le contrôle du son y échappent à l'emprise du temps, aux contraintes impérieuses du temps réel (…). C'est le temps différé qui permet d'étendre à la microstructure du sonore des possibilités de notation et de travail purement compositionnel ». Il ajoute : « On va vers une symbiose du temps réel et du temps différé : il faut réfléchir à ce qui peut et ce qui doit être commandé en temps réel » [Risset.93].

Dans notre cas, les simulations temps réel doivent permettre une situation de jeu instrumentale interactive et multisensorielle particulièrement sensible, mais il apparaît que cette dernière n'est pas systématiquement *souhaitable*[1].

Ces trois points montrent qu'au cours de son processus de création l'utilisateur voudra pouvoir choisir entre simulation temps réel et simulation temps différé et qu'en conséquence les deux modalités devraient cohabiter au sein de GENESIS. Le paradigme que nous considérons n'est donc plus la simulation interactive et multisensorielle d'objets physiques, mais plus simplement *la simulation d'objets physiques* qui, *dans certains cas*, sera effectivement interactive et multisensorielle.

Nous considérons ce point comme acquis pour la suite de l'analyse des tâches proposée dans cette partie. Qui plus, nous déciderons dans la suite de fonder GENESIS sur une approche exclusivement temps-différé, favorisant ainsi la simulation d'objets complexes et le processus de composition d'objets[2].

7.3 - Composantes structurelle et temporelle du son ; les deux temps de la création

La situation traditionnelle fonde toute démarche instrumentale sur deux phases successives : celle qui consiste à concevoir l'instrument (travail du luthier) et celle qui consiste à l'utiliser (travail de l'instrumentiste). Les sons engendrés par un instrument traditionnel procèdent de ces deux phases. Ils sont intimement structurés par une contribution *structurelle* et une contribution *évènementielle* :

> *«* La contribution structurelle c'est l'information qui est inscrite dans la structure de l'instrument, qui est par définition atemporelle (...). Et il y a l'autre contribution qui est la contribution temporelle du geste. (...) Le son est porteur du geste et en même temps il porte la structure de l'instrument » Claude Cadoz in [TableRonde.90a][3].

---

[1] Incidemment, on peut distinguer un autre intérêt très pragmatique au temps différé. Pour analyser par le canal visuel les évolutions des objets haute-fréquence, il est nécessaire de ralentir l'affichage. On pourrait, pour ce faire, réaliser un enregistrement complet de la simulation puis le repasser à une vitesse quelconque, en fonction de la précision d'observation que l'on souhaite. Cette solution n'est à ce jour pas raisonnable compte tenu de la quantité des données qu'il faudrait stocker. La solution consiste à ralentir le processus de simulation lui-même (i.e. : à simuler en temps-différé) et à afficher la position des modules au fur et à mesure de leur calcul.

[2] La connexion de GENESIS au simulateur temps réel sera toutefois évoquée ultérieurement comme étant une perspective à court terme pour la poursuite de nos travaux.

[3] En ce sens, *«* L'instrument est une classe de phénomènes sonores possibles, le geste instrumental détermine un évènement sonore singulier » [Cadoz.88b].

Ces deux phases essentielles dans la situation traditionnelle existent aussi avec GENESIS, comme l'a montré le parcours du chapitre 6. Créer avec CORDIS-ANIMA, c'est :

- d'une part *concevoir* l'objet simulé, responsable de la contribution structurelle et porteur d'un « potentiel sonore » ;
- d'autre part jouer sur cet objet, c'est-à-dire déterminer les signaux d'entrée responsables de la contribution temporelle. Pour ce faire, on peut avoir recours à une situation d'interaction temps réel (lorsqu'elle est possible) ou décider *d'éditer* ces signaux, de les *composer* préalablement à la simulation.

Nous appellerons la première étape *phase de lutherie* et la seconde *phase de jeu instrumental.*

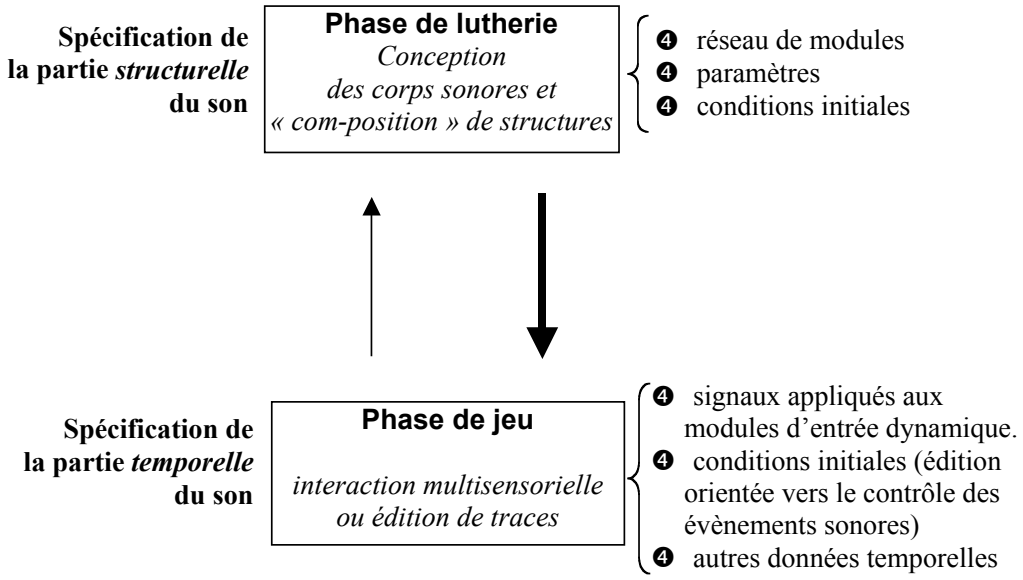


Figure 2 : les deux temps de la création du son

On peut considérer que les phases de lutherie et de jeu sont essentiellement successives au cours du processus de création avec GENESIS. Avec l'ordinateur il est cependant envisageable de passer de l'une à l'autre à tout instant.

Quoi qu'il en soit, au cours de chacune de ces phases l'utilisateur manipule des objets[1] différents et surtout effectue des tâches et a recours à des stratégies différentes. Pour cette raison, nous les envisagerons l'une après l'autre.

Dans la suite du document, nous déciderons de nous consacrer exclusivement au mode Lutherie et préciserons les raisons de cette décision.

[1] Au sens informatique ou de l'interaction homme-machine - et pas au sens "d'objet physique".

Chapitre 8

PHASE DE LUTHERIE : COMPOSITION D'OBJETS PHYSIQUES

La phase de lutherie consiste à concevoir un objet CORDIS-ANIMA afin qu'il puisse être simulé. Il s'agit donc *in fine* de définir un réseau de modules élémentaires, les paramètres physiques de chacun des modules et les conditions initiales des modules <MAT>.

Essentiellement, l'utilisateur peut adopter deux stratégies pour y parvenir :

- Une démarche *exploratoire*, qui consiste à mettre en œuvre librement le langage CORDIS-ANIMA. Avec une telle démarche, l'utilisateur s'intéresse avant tout aux phénomènes générés, notamment sonores et ne réfléchit pas nécessairement en termes physiques. Rappelons que les objets qui sont ainsi créés restent cependant des « modèles physiques » du fait de la *robustesse* du langage[1] et qu'on peut espérer obtenir nombre des avantages que nous avons décrits dans le chapitre 2.
- Un processus de modélisation, qui consiste à exprimer dans le langage CORDIS-ANIMA des *métaphores* d'objets ou « référents » réels afin non pas de reproduire la réalité mais de s'en inspirer. La démarche essentielle est ici une analyse structurelle[2].

Pendant la phase de lutherie, loin d'être opposées, ces stratégies pourront profitablement cohabiter et se compléter. Leur mise en œuvre repose sur un certain nombre de tâches et d'étapes conduites par l'utilisateur. Nous présentons ici successivement (voir également la figure 11 page 36) :

- Les opérations élémentaires qu'on peut déduire directement du langage CORDIS-ANIMA ;
- Les raisons pour lesquelles l'utilisateur a recours à la simulation en phase de lutherie ;
- Les stratégies itératives et incrémentales qu'il met en œuvre ;
- La notion de hiérarchie dans les objets ;
- Les notions d'échelle et leur importance ;
- Les différents cas où l'utilisateur met en œuvre une pensée macro-modulaire ;
- Le processus d'exploration des paramètres ;
- La notion de relation et de changement de base de paramètres ;
- Les deux catégories d'outils experts que sont *l'analyse* et la *génération* ;
- Les tâches d'ajout de commentaires et de légendes ;
- Différentes attitudes qui peuvent terminer la phase de lutherie.

[1] Le comportement d'un modèle CORDIS-ANIMA quelconque reste " plausible°" dans la mesure ou il émerge du comportement physique des modules élémentaires qui le composent. Voir partie I chapitre 2 et l'annexe B.

[2] Il s'agit de distinguer les différents constituants élémentaires de l'objet réel, de mettre en évidence les interactions entre ces constituants, puis de modéliser chacun d'eux. Cette démarche est récursive : elle peut s'appliquer à nouveau à chacun des constituants. Voir l'annexe B.

8.1 - Tâches élémentaires

Un choix fondamental pour nos travaux consiste à laisser à l'utilisateur la maîtrise complète de la « matière » constitutive des objets CORDIS-ANIMA que sont les modules, favorisant un processus *d'assemblage* des objets par opposition à un processus de découpage ou discrétisation d'un objet supposé continu.

Dès lors, *la phase de lutherie est fondée sur la prise en main du langage CORDIS-ANIMA* par l'utilisateur. On peut dresser la liste des opérations élémentaires que l'utilisateur est conduit à effectuer :

- Ajouter ou supprimer des modules ;
- Connecter les points L des <LIA> aux points M des <MAT> et, le cas échéant, déplacer ces connexions ;
- Affecter des valeurs aux paramètres des modules ;
- Définir les conditions initiales (position et vitesse initiales de chacun des modules <MAT>) afin soit de placer l'objet dans un état d'équilibre particulier au début de la simulation soit de l'exciter en introduisant une énergie cinétique ou potentielle non nulle.

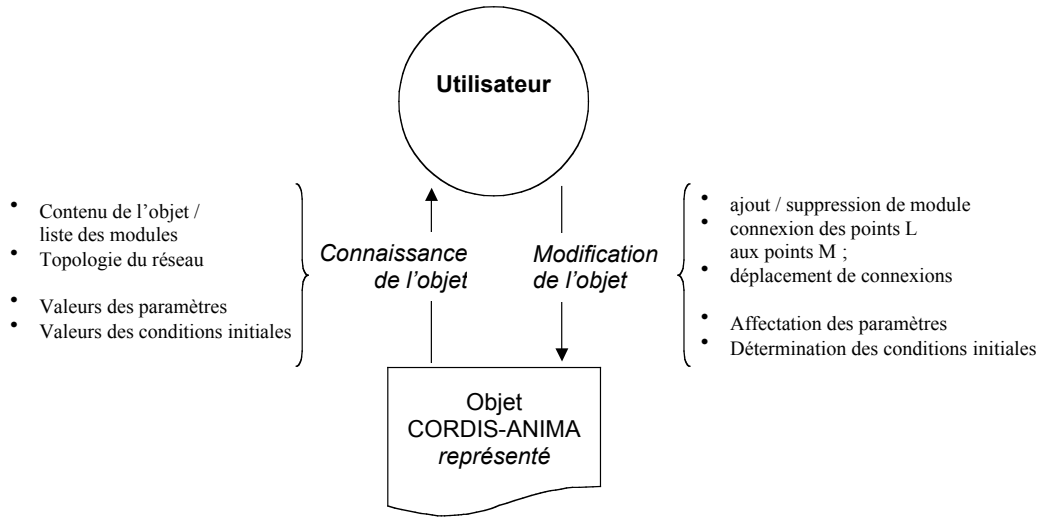


Figure 3 : tâches fondamentales de la phase de lutherie.

Une dernière tâche élémentaire essentielle à la phase de lutherie consiste à *prendre connaissance* de l'objet en cours de conception. Dans la mesure où le travail de l'utilisateur est fondé sur la maîtrise des modules, il s'agit ici essentiellement de s'informer de la structure du réseau <MAT>/<LIA> qui constitue l'objet, des paramètres et des conditions initiales de ses modules.

8.2 - La simulation en phase de lutherie

Lorsque l'utilisateur a recours à la simulation pendant la phase de lutherie, il ne cherche pas encore à *jouer* sur – ou avec – l'objet qu'il conçoit. On peut, en fait, distinguer quatre objectifs conduisant l'utilisateur à simuler l'objet pendant sa conception.

Évaluer

La perception des phénomènes engendrés lors de sa simulation demeure le principal critère d'évaluation de l'objet conçu en phase de lutherie[1].

Notons que si la simulation est à temps différé, l'évaluation sera essentiellement sonore[2]. Si elle est en temps réel, l'évaluation pourra être multisensorielle [3] et, plus important encore, interactive : action et perception seront intimement mêlées pour révéler les propriétés de l'objet.

Comprendre

La simulation apporte également un aide à la compréhension du comportement de l'objet. Il s'agit alors :

- Soit de *visualiser* l'état de l'objet au cours du temps ;
- Soit de procéder à des *mesures*. La situation de simulation offre la possibilité de *mesurer* les forces et positions échangées par les modules à tout instant. En analysant ces mesures, l'utilisateur peut mieux comprendre certains des phénomènes dont l'objet est le siège.

Régler les paramètres ; accordage dynamique

A supposer qu'il soit possible de modifier un objet – ou à minima ses paramètres – pendant qu'il est simulé en temps réel, l'utilisateur voudra utiliser la simulation pour faire converger *dynamiquement* et *à l'aide de la perception* les phénomènes générés vers ceux qu'il voudrait obtenir (accordage dynamique).

Obtenir un objet sonore à vocation musicale

Enfin, la simulation en phase de lutherie peut être utilisée pour générer non pas des phénomènes à évaluer, mais un objet sonore à vocation musicale sans qu'une phase de jeu n'intervienne. Il arrivera qu'au cours de l'une des démarches précédentes l'un des sons générés s'avère intéressant. Dès lors, l'utilisateur souhaitera disposer de cet *objet sonore* (au sens de la démarche concrète, donc notamment coupé de sa cause).

---

[1] Notons qu'un parallèle est possible avec l'attitude du luthier. Qu'il tape sur une caisse de résonance avant de l'assembler, essaye la mécanique de la clarinette, évalue au toucher le système d'échappement du piano ou encore joue quelques notes, c'est à ses *sens* qu'il confie en dernière analyse le soin d'évaluer l'instrument qu'il conçoit.

[2] L'écoute se rapproche de l'écoute "concrète" : l'utilisateur s'intéresse essentiellement à l'objet sonore et à son potentiel musical en faisant abstraction de l'objet CORDIS-ANIMA qui l'a généré.

[3] Il s'agit alors d'un jeu restreint, orienté non pas vers la création mais vers l'évaluation.

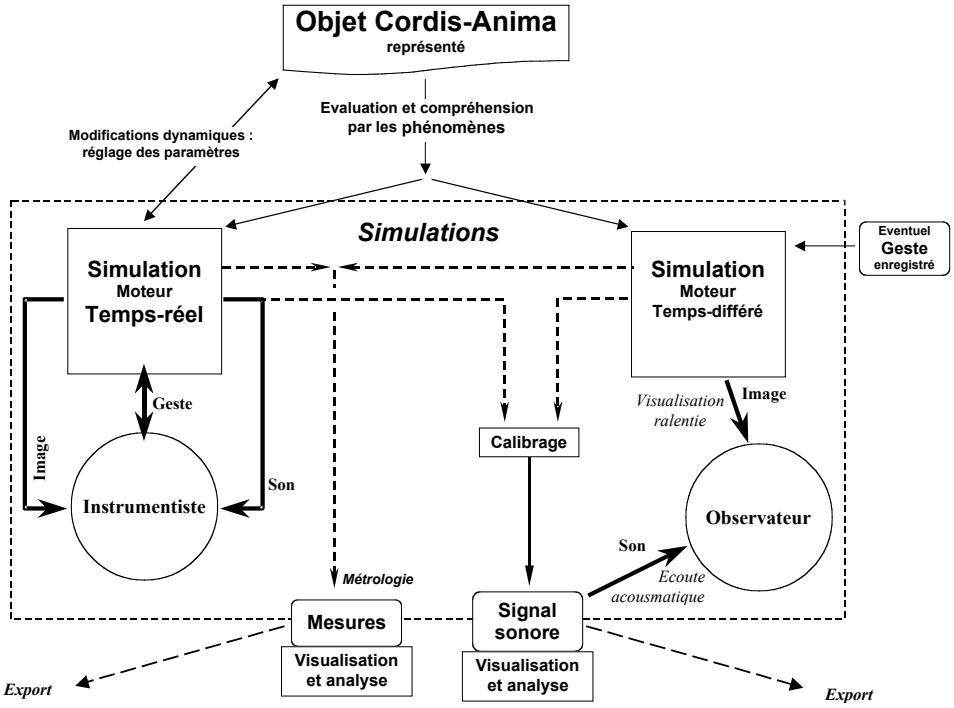


Figure 4 : la simulation en mode lutherie. Usages et fonctionnalités

La simulation et la perception des phénomènes qu'elle génère sont en conclusion essentiels à la phase de lutherie. La *stratégie incrémentale* que nous présentons ci-dessous s'organise, d'ailleurs, autour de la simulation.

8.3 - Stratégies itératives et incrémentales

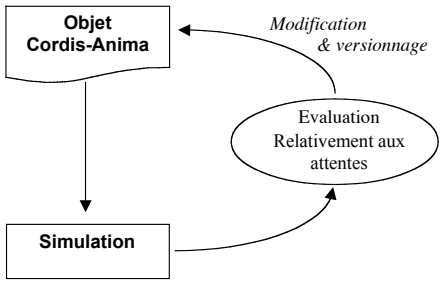


Figure 5 : démarche itérative en mode lutherie : la boucle modification – simulation - évaluation

L'utilisateur, en général, n'obtient pas immédiatement un objet qui le satisfait. Il a au contraire recours à une stratégie itérative (figure 5) [1] : Il modifie l'objet ;

- Il l'évalue – notamment en le simulant.
- Il détermine, compte tenu des résultats obtenus, les modifications prochaines.

Une autre caractéristique essentielle de la démarche de l'utilisateur en phase de lutherie est la stratégie incrémentale. Elle s'analyse en trois points :

[1] *Tous* les utilisateurs des versions successives du logiciel ont eu recours à une telle stratégie. Notons que la démarche itérative n'est pas spécifique à GENESIS. Vinet observe au contraire qu'elle est très générale lorsque l'ordinateur est utilisé comme un outil de création : « Ainsi, par une succession de cycles rapides d'essais / erreurs combinant commande et évaluation à l'écoute du son produit, l'utilisateur peut converger vers le résultat souhaité ou décider à un moment donné de détourner son objectif pour suivre les pistes qui se manifestent de manière inattendue » [Vinet.99a].

- Les objets sont complexifiés progressivement par l'adjonction de nouvelles parties ;
- Ces parties, souvent, sont travaillées séparément avant d'être insérées dans l'objet complet.
- L'utilisateur constitue progressivement un *catalogue de structures* (figure 6) qui rend compte de son expertise sous la forme d'objets réutilisables. En le développant, il incrémente son savoir-faire.

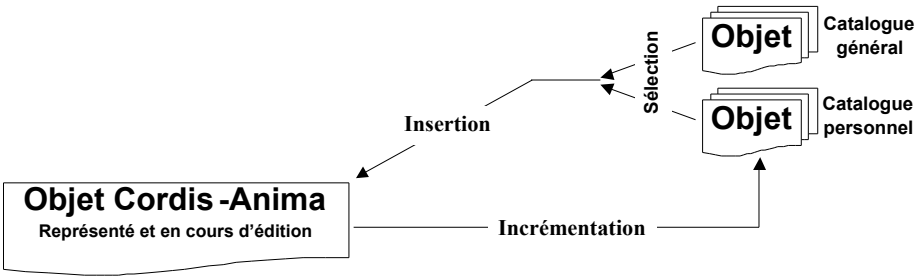


Figure 6 : catalogue, insertion de structures.

La figure précédente fait apparaître un *catalogue général*. Un tel catalogue est constitué progressivement au sein du laboratoire depuis plusieurs années. Les utilisateurs y font régulièrement appel.

8.4 - Conception hiérarchique

L'analyse structurelle des scènes instrumentales traditionnelles montre l'existence d'une chaîne instrumentale récurrente. On distingue alors en général : l'excitateur (ou tout autre objet intermédiaire entre l'instrumentiste et l'instrument), la structure vibrante, le chevalet, l'environnement local (résonateur par exemple) et l'environnement global (milieu de propagation du son, salle, etc.).

Lorsqu'il met en œuvre la stratégie de modélisation, l'utilisateur est naturellement conduit à proposer des métaphores pour chacun de ces constituants et, en conséquence, à introduire en aval de la structure vibrante une *hiérarchie* entre différentes parties des objets qu'il conçoit.

Le chapitre précédent a montré qu'il est possible d'étendre cette structuration hiérarchique en amont des structures vibrantes, en ajoutant des objets basse-fréquence à la scène. Ces objets ont pour fonction la génération d'*évènements*, la structure vibrante étant, quant à elle, responsable du *timbre*. Ils constituent des *métaphores d'instrumentistes* ou plus généralement d'opérateurs humains, exprimées dans le langage CORDIS-ANIMA.

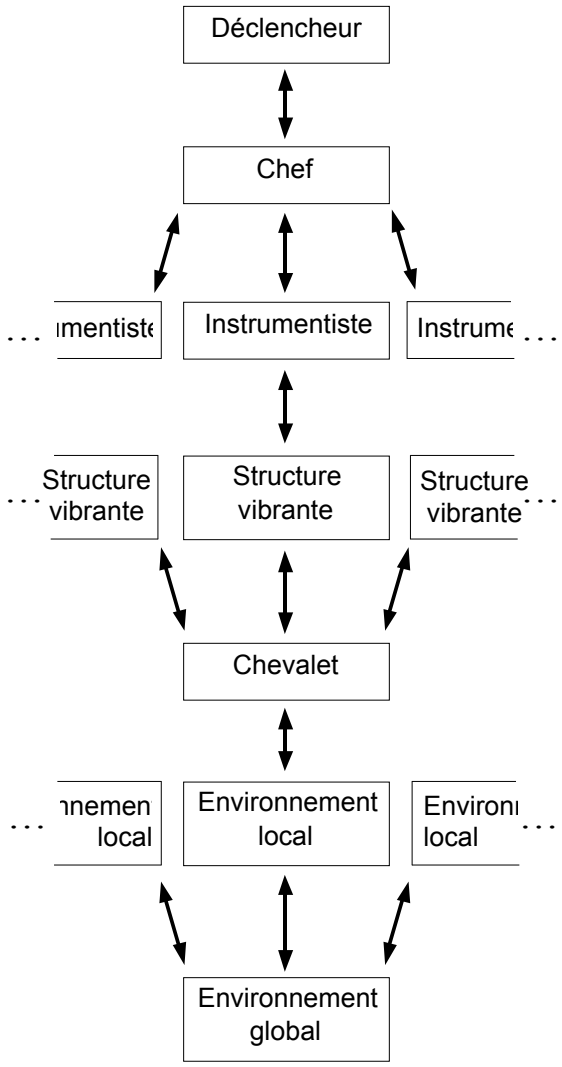


Figure 7 :
La chaîne instrumentale augmentée au sein des objets CORDIS-ANIMA.

Pour caractériser la démarche hiérarchique propre au processus de « composition » d'objets, nous introduisons donc la notion de *chaîne instrumentale augmentée* (ci-contre). Y figurent à la fois des objets « générateurs de timbre » et des objets « générateurs de forme ». Plus précisément, chacun des maillons est défini par sa *fonction* dans la hiérarchie des objets. Par exemple :

- Le chef dirige l'instrumentiste ;
- L'excitateur réalise la médiation entre l'instrumentiste et la structure vibrante ;
- La structure vibrante est l'objet où naît la vibration à fréquence audible ;
- Le chevalet permet à la fois la sommation des phénomènes générés dans plusieurs structures vibrantes et la mise en interaction de ces structures (il permet l'existence d'un phénomène de vibration par sympathie) ;
- L'environnement global permet la sommation de plusieurs instruments, sans pour autant les mettre en interaction forte ; à son niveau, en outre, on peut envisager de travailler un éventuel « effet de salle » ;
- Etc.

A cette liste s'ajoutent les *déclencheurs*, objets dont la structure et la fonction sont quelque peu particulières et nécessitent quelques précisions.

Un déclencheur est constitué d'un MAS et d'une liaison non-linéaire qui développe une force nulle tant que les deux <MAT> connectés sont éloignés – typiquement : un <LIA> de type BUT. La fonction d'un déclencheur est de modifier par une interaction physique, un *corps à corps*, l'état énergétique de la structure qu'il « déclenche ».

On peut avec un déclencheur contrôler à la fois la nature (en choisissant le type et les paramètres de la liaison non linéaire), l'intensité (avec la vitesse initiale ou « d'incidence » du MAS) et la date (par la position initiale du MAS, dans la mesure ou cette position va déterminer la durée de la phase d'incidence) de l'événement généré. Si l'on omet les éventuels modules d'interaction (voir plus loin), les déclencheurs sont en général les principales sources d'énergie au sein d'un objet composé.

Aucun des maillons de la chaîne instrumentale augmentée ne prend le pas sur les autres. L'utilisateur se penche successivement sur chacun d'entre eux, mais il le fait dans un ordre quelconque suivant qu'il préfère travailler d'abord à l'échelle des processus sonores ou à celles des processus générateurs d'évènements.

Enfin, notons que si elle a été souvent mise en œuvre dans des situations de création, la hiérarchie proposée ne reste qu'un exemple. A part, peut-être, la structure vibrante (dans la mesure ou l'objectif reste la génération de son), certains maillons peuvent être omis et d'autres imaginés ou ajoutés. Nous retiendrons donc l'essentiel : un objet CORDIS-ANIMA est un objet structuré et hiérarchisé.

8.5 - Les notions d'échelle

La hiérarchisation suffit à montrer que plusieurs « échelles » doivent être manipulées par l'utilisateur : un chef, un instrumentiste, une structure vibrante ou un chevalet ne sont pas à la même échelle. Pour autant, l'échelle d'un objet ne s'identifie pas à sa fonction[1]. La notion d'échelle peut être envisagée sous deux angles :

- Par les plages des paramètres physiques et/ou conditions initiales caractéristiques de l'échelle considérée. Par exemple, un *chef* se caractérise par certaines plages de valeurs de l'inertie, de la raideur, etc.
- Par les caractéristiques des phénomènes générés par les objets de l'échelle considérée. Par exemple, un instrumentiste est une structure basse fréquence, dont les mouvements s'amortissent sur un temps relativement long et dont l'amplitude des mouvements est grande.

Nous adopterons ce dernier point de vue pour mettre en évidence les trois *échelles* dont la maîtrise durant la phase de lutherie revêt une importance considérable.

Echelle d'amplitude

La première échelle est celle de l'amplitude des phénomènes générés. Pour qu'un objet CORDIS-ANIMA complexe se comporte de façon satisfaisante, l'utilisateur doit, tôt ou tard, s'intéresser aux amplitudes relatives des phénomènes dynamiques propres à ses différentes parties[2].

Echelle des fréquences et constantes de temps

La seconde échelle essentielle est relative aux constantes de temps de l'objet. Elle adresse les notions de fréquence et de *temps d'amortissement*. Pour la fréquence, on distingue notamment : les objets sans fréquence propre (ou sur-amortis), les objets à fréquence audio (typiquement les structures vibrantes) les objets à fréquence gestuelle (de l'ordre du Hertz) les objets à très basse fréquence. On comprendra qu'une certaine maîtrise de l'échelle des fréquences (ou, réciproquement, des périodes) par l'utilisateur soit nécessaire dans le contexte de la création musicale.

[1] Par exemple, pour être efficace, un déclencheur devra avoir une inertie très différente suivant qu'il est connecté à un chef ou à une structure vibrante. De même, il peut arriver qu'un instrumentiste et une structure vibrante soient à la même échelle d'amplitude, etc.

[2] Si par exemple il souhaite « écouter » sur une voie sonore le mouvement d'une structure vibrante et sur une seconde celui d'une autre, il est essentiel que les amplitudes de leurs mouvements soient comparables – sans quoi l'une d'entre elle ne sera pas audible.

Echelle d'impédance

La troisième échelle concerne l'inertie des objets ou, pour être plus précis, leur impédance relative. Cette échelle détermine l'importance des échanges énergétiques ou encore la façon dont deux parties d'un objet complexe, mises en interaction, vont collaborer et s'influencer mutuellement. Elle permet par exemple de choisir la façon dont un instrumentiste CORDIS-ANIMA sera influencé par l'instrument avec lequel il interagit. La maîtrise de l'échelle d'impédance par l'utilisateur est donc essentielle.

Aux notions d'échelles sont associées les tâches de *mise à l'échelle*, fréquentes au cours de la phase de lutherie. De façon générale, apparaît la nécessité d'une maîtrise des échelles et des interactions entre objets d'échelles différentes.

8.6 - La macro-modularité dans le processus de lutherie

L'organisation hiérarchique que nous avons décrite et le fait qu'au sein d'un objet cohabitent souvent un grand nombre de modules[1] suffisent à montrer que l'utilisateur doit parfois s'abstraire du *niveau des modules* pour considérer des ensembles cohérents. La démarche macro-modulaire est en fait très générale au sein de GENESIS et d'autres raisons peuvent amener l'utilisateur à la mettre en œuvre.

Nous distinguons d'abord la macro-modularité *fonctionnelle* et la macro-modularité *morphologique*.

Macro-modularité fonctionnelle ; sous objet

Lorsqu'il conçoit ou manipule un objet complexe, l'utilisateur ne le considère pas comme une collection de modules élémentaires CORDIS-ANIMA, mais comme l'assemblage de plusieurs parties.

On dira alors qu'un ensemble *fonctionnel* ou *sous-objet* est constitué lorsque que l'utilisateur a décidé qu'il a une *fonction* dans l'objet global[2]. Un ensemble fonctionnel peut être, par exemple, une métaphore de corde, un instrument constitué de plusieurs cordes et de chevalets, un instrumentiste, l'ensemble de ces derniers, etc.

Macro-modularité morphologique

On dira par symétrie qu'un ensemble *morphologique* est constitué dès lors que l'utilisateur veut considérer globalement certaines propriétés de ses modules – par exemple leurs paramètres – sans que l'ensemble ne soit nécessairement un sous-objet. Les MAS homogènes d'un sous-objet peuvent par exemple constituer un ensemble morphologique, de même que plusieurs modules appartenant à deux sous-objets mais qui, pour une raison ou pour une autre, doivent être considérés ensembles.

Un autre caractère de la démarche macro-modulaire tient à la *permanence* des ensembles constitués. Nous distinguerons alors les ensembles *forts* des ensembles *faibles*.

[1] Souvent jusqu'à plusieurs milliers.

[2] Notons toutefois que la macro-modularité fonctionnelle ne s'identifie pas forcément à la structuration hiérarchique de l'objet. C'est, en fait, une notion plus large.

Ensemble faible

Nous dirons qu'un ensemble est *faible* lorsqu'il est nécessaire à la réalisation d'une tâche particulière et *momentanée* mais que l'utilisateur souhaite conserver par ailleurs un accès individuel aux modules qui le constituent. Un ensemble faible est par exemple créé lorsque l'utilisateur veut modifier conjointement les paramètres de plusieurs modules homogènes mais conserver la possibilité d'en modifier le réseau[1].

Ensemble fort ; macro-modules

Nous dirons que l'utilisateur a créé un ensemble *fort* ou *macro-module* dès lors qu'il souhaite envisager globalement certains modules sans plus avoir accès à chacun d'entre eux. Les paramètres pertinents pour un tel macro-module ne coïncident pas toujours avec la liste des paramètres de tous les modules qui le composent. L'utilisateur voudra donc définir des paramètres spécifiques au macro-module et les répercuter ensuite sur les paramètres CORDIS-ANIMA.

Le fait d'identifier un ensemble fort limite la *variabilité* à la fois en termes structurels (la structure du réseau <MAT>/<LIA> de l'ensemble est figée) et en termes de paramètres (certaines variations paramétriques sont privilégiées). Du même coup cependant, les ensembles forts permettent de restreindre la difficulté de compréhension et de manipulation – et c'est là probablement l'un de leurs intérêts majeurs.

La démarche macro-modulaire est portée en puissance par CORDIS-ANIMA, langage fortement modulaire, mais n'est pas pris en compte directement par le formalisme. C'est au niveau de l'environnement GENESIS que les fonctionnalités permettant de la soutenir devront être entièrement conçues.

8.7 - Détermination des paramètres ; exploration paramétrique

Pour effectuer le réglage des paramètres, l'utilisateur doit disposer d'une bonne connaissance de l'espace des paramètres. Il faut notamment qu'il ait une idée des plages valides pour chaque paramètre (compte tenu des *échelles* auxquelles il souhaite travailler). Il faut aussi qu'il ait pu identifier des tendances entre les paramètres CORDIS-ANIMA et les phénomènes ou, plus encore, entre ces paramètres et certaines dimensions de la perception. Certains effets des paramètres sont assez aisés à repérer, tel l'incidence de la *raideur* sur la *fréquence*. D'autres sont plus complexes.

En la matière, le savoir-faire de l'utilisateur, acquis au fur et à mesure de sa confrontation à CORDIS-ANIMA, est sans doute particulièrement important. Il s'avère insuffisant dans de nombreux cas.

[1] Il peut cependant être intéressant de conserver un ensemble faible d'une manière ou d'une autre lorsque la tâche concernée doit être répétée même s'il n'a pas de raison d'être hors du contexte de cette tâche.

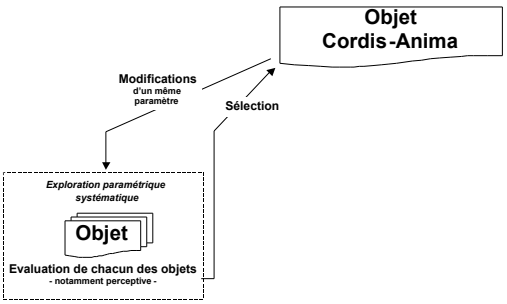


Figure 8 : exploration paramétrique systématique

Une démarche d'exploration systématique des effets de certains paramètres est alors nécessaire. Nous l'avons évoquée à plusieurs reprises au cours du parcours du chapitre 6 et les utilisateurs des versions successives de GENESIS l'ont, de fait, mise en œuvre de façon récurrente. Elle consiste à faire varier quelques paramètres selon une grille d'exploration afin d'obtenir une « image phénoménologique » de l'espace des paramètres.

8.8 - Relations et changement de base de paramètres

Relation entre paramètres et phénomènes ou propriétés

Une exploration paramétrique poussée conduit dans certains cas optimaux à identifier une *relation algébrique* liant les paramètres CORDIS-ANIMA à certaines propriétés quantifiables symbolisées par un meta-paramètre.

Quelle que soit la démarche pour l'obtenir, une telle relation offre plusieurs intérêts. A minima, indiquons qu'elle permet d'obtenir par le calcul une mesure de la propriété à partir des paramètres de l'objet, et réciproquement, de déterminer les paramètres CORDIS-ANIMA qui assurent l'obtention d'une propriété souhaitée – par exemple la fréquence fondamentale d'un ensemble de modules, la qualité d'une percussion ou encore l'amortissement critique d'un oscillateur.

Changer la base des paramètres

En définissant des relations entre paramètres CORDIS-ANIMA et meta-paramètres, l'utilisateur peut également changer la *base de paramètres* de son modèle.

Ces relations de changement de base peuvent être définies de façon arbitraire. L'essentiel est qu'en contrôlant les meta-paramètres l'utilisateur force les paramètres CORDIS-ANIMA à évoluer sur certaines surfaces. Certains comportements sont alors privilégiés, ainsi que certaines dimensions de la perception. On comprendra qu'un tel changement de base est tout à fait intéressant en situation de création.

8.9 - Expertise, analyse et génération : vers un contrôle explicite des propriétés

Certains procédés d'analyse physique et mathématique systématiques permettent d'établir de façon automatisée des « ponts » entre l'univers des objets CORDIS-ANIMA et celui de leurs propriétés.

Il convient de rappeler ici que l'un des intérêts majeurs de CORDIS-ANIMA tient au fait que l'utilisateur travaille au niveau physique (les causes), en *pensant physique*, et non pas au niveau du signal ou des propriétés (les effets). Le recours à ce type de procédés automatisés n'est donc pas essentiel dans la démarche de l'utilisateur. Il peut toutefois apporter des solutions aux difficultés qu'il rencontre dans la conception des objets.

Ces outils se répartissent en deux catégories : ceux qui réalisent une analyse d'une part, ceux qui permettent une génération d'autre part (figure 9).

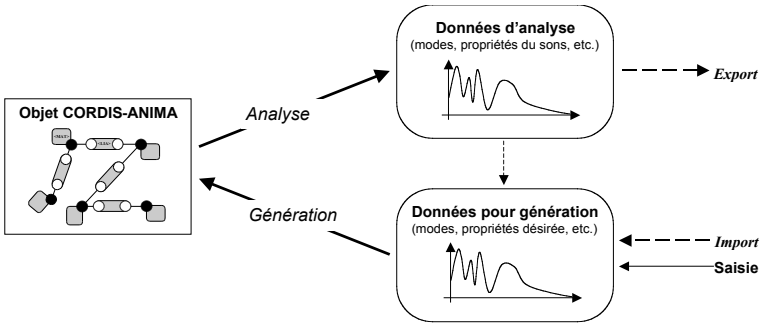


Figure 9 : Principes de l'analyse et de la génération

L'analyse

L'analyse consiste à déterminer de façon automatisée certaines des propriétés d'un objet donné sans avoir recours à la simulation. Par exemple, l'analyse modale offre la connaissance des modes propres de l'objet et l'analyse des divergences permet de repérer les causes possibles d'une divergence numérique du « modèle » CORDIS-ANIMA. Notons que l'existence de procédés d'analyse laisse entrevoir une ouverture de GENESIS par *l'exportation* éventuelle des données résultantes vers d'autres logiciels et/ou machines.

La génération d'objets ou de paramètres

La génération est réciproque de l'analyse. Elle peut concerner différents niveaux dans les objets, suivant qu'il s'agit de générer *un réseau* CORDIS-ANIMA complet et paramétré ou de façon plus restreinte les *paramètres* d'un objet dont on aurait préalablement choisi la structure. La génération repose en outre sur deux catégories de données :

- Les *contraintes* imposées à l'objet, sa structure ou ses paramètres (taille ou topologie de l'objet à générer, conservation de certains paramètres ou des relations de proportionnalité entre paramètres, homogénéité, etc.). Elles sont nécessaires pour l'élimination de certaines solutions et conditionnent la précision de la correspondance possible entre propriétés de l'objet *généré* et propriétés désirées.
- Les *données qui spécifient les propriétés*. Elles représentent une possible ouverture de GENESIS vers d'autres environnements via *l'importation* de ces données. Notons qu'un cas particulier mais particulièrement difficile à traiter est celui ou un signal (un son ou une trace gestuelle par exemple) est importé puis « traduit en langage CORDIS-ANIMA » sous la forme d'un objet qui en permet la re-synthèse.

8.10 - Commentaires, légendes

En plus de l'ensemble des tâches relevant de la conception de l'objet, l'utilisateur voudra *commenter*, annoter, légender son travail. Ces informations diverses n'ont aucun effet sur la simulation elle-même mais permettent :

- De reprendre ultérieurement un travail commencé ;
- De le communiquer plus aisément à d'autres personnes.
- D'ajouter, plus généralement, des informations symboliques qui surchargent l'objet sans en modifier le comportement.

8.11 - Différentes sorties de la phase de lutherie

Plusieurs voies peuvent être choisies pour mettre un terme à la phase de lutherie (voir figure 10).

Archivage

La constitution d'un catalogue d'objet est l'un des aspects de la démarche incrémentale. Elle nécessite :

- L'ajout de commentaires et légendes aux objets et aux différents sous-objets qui les constituent ;
- Une stratégie d'archivage de la part de l'utilisateur et, pour la mettre en œuvre, un certain nombre de tâches ;
- Des fonctionnalités pour permettre une navigation aisée dans le catalogue d'objets et les sons qui leur correspondent.

Exportation d'objets sonores

Dès lors que l'utilisateur considère les sons obtenus lors des simulations comme musicalement pertinents, il peut souhaiter les exporter. Un son ainsi exporté acquiert le statut d'objet sonore (au sens de la démarche concrète) ; il est coupé de sa cause.

Constitution d'une famille instrumentale

Une autre démarche terminale consiste à procéder à des *variations* successives autour de l'objet qui vient d'être conçu et à archiver les objets et les sons obtenus. Cette façon de procéder tire partie de la *robustesse* du formalisme CORDIS-ANIMA[1] : en modifiant certains paramètres ou en transformant légèrement la structure de l'instrument l'utilisateur a des chances d'obtenir des variations musicalement pertinentes tout en conservant une certaine permanence de timbre. La démarche aboutit ainsi à la définition d'une *famille instrumentale*, c'est-à-dire d'un ensemble d'objets partageant certaines propriétés structurelles et paramétriques, ainsi que certaines propriétés phénoménologiques.

Préparation pour le jeu

Enfin, une quatrième voie consiste à préparer l'objet pour le jeu. Il s'agit alors, essentiellement, d'ajouter un ou plusieurs modules d'interaction, c'est à dire de déterminer les points d'accès à l'objet. On entre ensuite dans la seconde phase fondamentale de la création avec la simulation d'objets physiques : la phase de jeu. Nous lui consacrons le chapitre suivant.

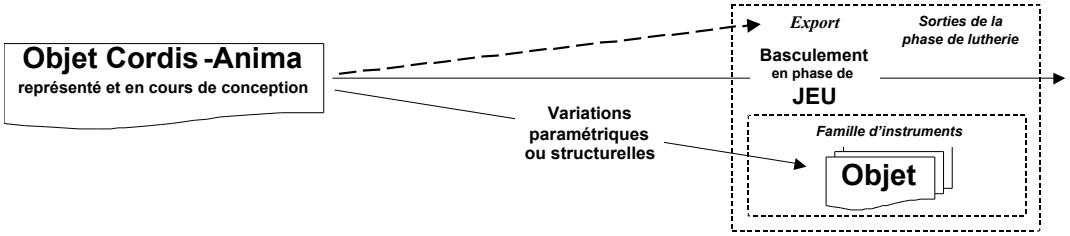


Figure 10 : sorties possibles de la phase de lutherie

[1] Voir partie I paragraphe 2.6 et l'annexe B.

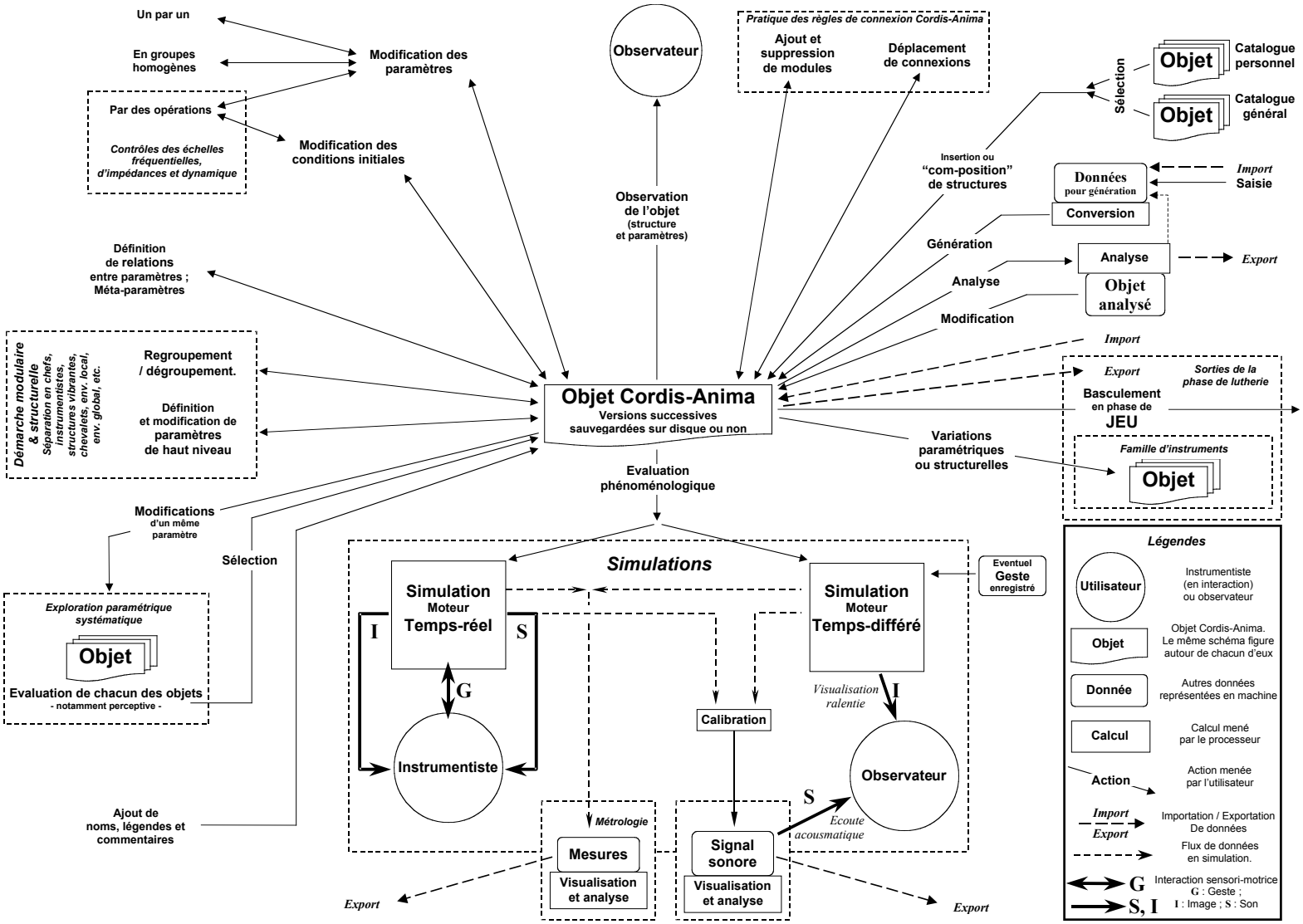


Figure 11 : récapitulatifs des tâches fondamentales impliquées dans la phase de Lutherie

Chapitre 9

PHASE DE JEU

ET AUTRES ASPECTS DE LA CREATION AVEC GENESIS

La phase de jeu a pour objet de définir la *contribution temporelle* qui isolera un son particulier parmi l'ensemble de ceux que peut générer un objet préalablement conçu. Pendant la phase de jeu, l'utilisateur s'intéresse (implicitement ou explicitement) à des données qui s'inscrivent dans le temps, c'est à dire à des signaux. Il s'agit plus précisément de spécifier :

- Les *traces gestuelles*, c'est à dire les signaux de force et de position entrant dans la simulation par les modules d'interaction placés à certains endroits stratégiques de l'objet. Pour cela, l'utilisateur peut soit avoir recours à une simulation interactive et multisensorielle, soit éditer les traces gestuelles préalablement à la simulation.
- Les *conditions initiales* des modules de l'objet qui vont déterminer à la fois la qualité et la date de certains évènements. Incidemment, on peut considérer les conditions initiales comme des « gestes » élémentaires qui seraient appliqués, au début de la simulation, à chacun des <MAT> de la scène[1].

L'identification des tâches impliquées dans la phase de jeu pose des problèmes particuliers. En effet :

- L'utilisateur, en phase de jeu, n'est pas contraint par un formalisme qui pourrait nous servir de base ;
- Nous n'avons pas pu observer d'utilisateur en phase de jeu et nourrir notre réflexion de cette observation pour la bonne raison que les outils qui devront la soutenir n'existent pas encore. Notamment, le moteur temps réel TELLURIS de conception très récente ne peut aujourd'hui être mis en œuvre que dans des conditions expérimentales peu satisfaisantes dans une perspective de création ;

Par ailleurs, dans la mesure où nous déciderons plus loin de nous consacrer essentiellement à la phase de lutherie, l'analyse des tâches propres à la phase de jeu revêt, dans ce document, une importance moindre que ce qui précède. Nous proposons cependant quelques résultats préliminaires.

Dans ce chapitre, nous évoquerons successivement :

- La situation d'interaction multisensorielle *via* le TGR© ;
- D'autres flux de données envisageables en cours de simulation ;
- Les tâches d'édition de la trace et la notion de représentation causale de la trace ;
- L'existence, enfin, de nombreuses autres pistes. Sans être détaillées, elles montrent que le voie qui mènera à une version finalisée de GENESIS est encore longue.

[1] Les conditions initiales des MAT (position et vitesse initiales) sont équivalentes à des signaux d'une durée de 2 échantillons (position et position retardée de chacune des MAS).

9.1 - Jouer avec l'instrument ; interaction multisensorielle via le TGR©

En phase de jeu, il s'agit d'abord de jouer en temps réel avec l'objet via le TGR©, bénéficiant ainsi la de la qualité de la relation instrumentale permise par l'interaction multisensorielle et particulièrement le retour d'effort[1].

L'obtention d'une telle situation est conditionnée par deux points :

- Des modules d'interaction, image dans l'objet des touches du TGR©, doivent avoir été préalablement ajoutés pendant la phase de lutherie.
- L'objet simulé doit être suffisamment simple pour pouvoir être simulé en temps-réel. L'interaction multisensorielle n'est pas notamment possible avec les objets complexes qui résultent d'un processus de « composition » poussé. Dans ce cas, la phase de jeu (qui d'ailleurs n'est pas indispensable) doit nécessairement reposer sur d'autres procédés.

La situation de jeu temps réel est l'un des pôles essentiels de la phase de jeu. Elle est aussi à l'image de la situation instrumentale traditionnelle de sorte qu'il n'est pas nécessaire de la commenter plus avant.

9.2 - D'autres flux de données

Au paragraphe 7.1 - nous avons indiqué qu'il était nécessaire *d'ouvrir* GENESIS pour qu'il soit à même de communiquer avec d'autres machines du studio audionumérique. On peut avancer que parmi les données d'import/export figurent en bonne place les *flux dynamiques* (i.e. : pris en compte pendant la simulation) que pourraient échanger GENESIS avec des logiciels de synthèse et de traitement ou des périphériques.

Une première piste consiste à choisir un *son* comme signal de commande d'un module d'entrée. GENESIS devient alors un logiciel de traitement des sons par modèle physique.

Une seconde piste, que nous considérons comme essentielle, consiste à intégrer les standards de communication entres machines numériques musicales qui émergent actuellement par suite des nombreuses recherches menées sur le contrôle temps réel. Ce qui est en puissance ici, c'est en fait la création d'un pont entre deux univers très différents avec :

D'un coté	*De l'autre*
La simulation physique	Les techniques signal
La « pensée physique »	Le vocabulaire du signal
Le geste *interactif*	Le geste de *commande* et de *contrôle*
La trace gestuelle, signal *échantillonné*	Les signaux *évènementiels*

[1] Rappelons que la situation d'interaction multisensorielle peut être aussi utile en phase de lutherie. Il ne s'agit cependant pas alors de jouer avec l'objet mais plus de l'évaluer.

9.3 - Enregistrement et relecture de traces gestuelles

Parmi les tâches de l'utilisateur figurent :

- L'enregistrement d'une trace et donc son archivage dans un *catalogue de traces* ;
- L'incrémentation du jeu : pendant qu'une trace gestuelle préalablement enregistrée est ré-appliquée, l'utilisateur peut jouer sur une autre partie de l'objet.
- L'application d'une trace gestuelle lors d'une simulation ultérieure sur un objet différent, le cas échéant plus complexe et simulé en temps différé.

9.4 - Edition du signal trace : la boucle d'édition de la trace

La trace peut également être éditée. Nous évoquons dans ce paragraphe quelques-unes des opérations d'édition envisageables (figure 12).

Représentation et analyse

Editer la trace suppose qu'une représentation en soit proposée. La représentation la plus simple qu'on puisse envisager est celle du signal – à l'image de celle des éditeurs de sons. D'autres approches de la représentations des traces sont possibles : la trace peut envisagée en *référence à l'action* (i.e. : au geste réalisé) ou à la *perception* (i.e. : à l'effet qu'elle induit sur la perception auditive) [Ramstein.91].

Segmentation et concaténation

La segmentation et le montage de traces sont les transformations les plus simples que peut vouloir mener l'utilisateur. Il s'agit d'abord d'extraire d'une trace enregistrée certains segments pour constituer un catalogue – de telle sorte qu'à chaque segment corresponde à la fois un *geste réalisé* et un *effet.* Il s'agit, ensuite, de générer une nouvelle trace par la concaténation de plusieurs segments.

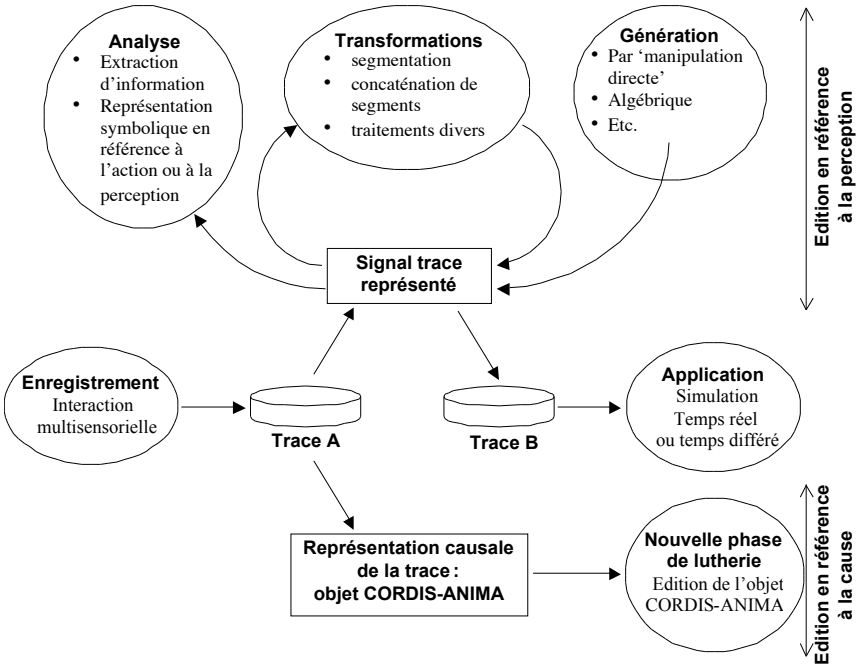


Figure 12 : Boucles pour l'edition de la trace

Traitements divers

Pour mettre en œuvre la boucle d'édition de la trace, l'utilisateur peut choisir entre deux démarches :

- Une démarche exploratoire inspirée de la démarche concrète : toute transformation est envisageable (renversement, étalement, filtrage par exemple) est ici envisageable pour peu que l'utilisateur dispose des outils de traitement du signal adéquats. L'essentiel est qu'elle soit validée par la perception. Rappelons toutefois qu'entre l'opération de transformation et la validation figure nécessairement une simulation.
- Une démarche basée sur une connaissance des opérations « légitimes » sur la trace ainsi que de leurs effets. Ce qui est jeu ici, c'est la conservation de l'adéquation de la trace à l'objet auquel elle est appliquée. Le savoir-faire en la matière reste aujourd'hui limité [Ramstein.91] et une recherche approfondie doit se poursuivre.

Quelle que soit la démarche adoptée, il est préférable que les opérations sur le « signal trace » soient menées avec précaution. Ce signal est en effet enregistré au cours d'une *interaction* et porte donc intimement le geste réalisé mais aussi l'histoire de l'objet. Par contre, lors de sa réutilisation, il sera *imposé* à l'objet. Dès lors, appliquer une trace transformée – ou même plus simplement segmentée – c'est risquer de ne pas obtenir un comportement *plausible*.

Edition causale de la trace

Une solution à ce problème réside peut-être dans le principe de l'édition *causale* de la trace.

Il s'agit alors d'obtenir un objet CORDIS-ANIMA qui simule l'instrumentiste et re-synthétise la trace enregistrée[1]. Dès lors qu'on dispose d'un tel objet, il est possible de mettre en œuvre une nouvelle phase de lutherie pour le transformer. Ce faisant, on « édite » comme précédemment la trace – c'est à dire qu'on modifie les évènements générés. En revanche, une interaction physique – simulée – a lieu. On peut espérer qu'elle garantira la cohérence et la *plausibilité* des phénomènes obtenus.

Génération

Enfin, notons que l'utilisateur peut vouloir générer un signal trace en le dessinant ou en utilisant une représentation algébrique par exemple.

Plusieurs expériences ont montré la pertinence d'une telle approche (voir [Castagne.96], [Castagne&Cadoz.00] par exemple). Deux remarques peuvent être faites :

- Tout d'abord, la génération de trace ne nécessite à aucun moment d'avoir recours à une simulation temps réel ; elle peut donc être mise en œuvre sur des objets complexes et en l'absence de tout système interactif.
- Ensuite, la génération de trace offre un contrôle explicite de la dimension temporelle, c'est-à-dire des instants ou les évènements doivent intervenir. C'est une particularité intéressante lorsqu'on compare le procédé à celui qui consiste à construire un *instrumentiste* CORDIS-ANIMA. Par contre le recours à une trace générée rompt à nouveau la situation d'interaction.

[1] Les procédés de génération d'objets (paragraphe 8.9 - ci-dessus) peuvent sans doute y aider.

9.5 - D'autres horizons

En phase de lutherie, la construction d'un *instrumentiste* permet d'une part de disposer d'un procédé générateur d'évènements, d'autre part d'obtenir une situation d'interaction entre instrumentiste et structure vibrante, assurant ainsi une bonne cohérence des comportements de l'objet et une bonne *plausibilité* des sons. Les conditions initiales, envisagées alors comme des gestes élémentaires, permettent de *déclencher* et des séries d'évènements sonores et de les contrôler en terme de *date* et de *caractère*, notamment d'intensité[1].

En phase de jeu, l'utilisateur s'attache à spécifier des données de nature temporelle. Il a recours à la situation d'interaction multisensorielle ou met en œuvre divers procédés d'édition de trace. Dans les deux cas, la dimension temporelle est explicite.

La coupure entre ces deux phases est, en fin de compte, du ressort de l'utilisateur.

Pour le moment, nous n'avons parlé que de la réalisation *d'un* objet, puis, sur cet objet, de la réalisation *d'un* geste.

Cependant, au sein d'une même composition, l'utilisateur souhaitera sans doute utiliser plusieurs objets et réaliser successivement plusieurs gestes sur chacun d'entre eux.

Il peut, certes, les réunir au sein d'une macro structure, dont les différentes parties seraient sollicitées successivement. On comprend cependant la limite d'une telle approche : elle impose la conception d'objets très complexes et ne prévoit pas, par exemple, la succession sur une même structure de phases de jeu interactif et à partir de traces.

Aussi le compositeur voudra-t-il sans doute dans son processus de création prendre de la distance d'une part avec les objets CORDIS-ANIMA, d'autre part avec les gestes ou leur réalisation. Une troisième phase du processus de création qui opèrerait dans un domaine plus abstrait ou symbolique pourrait être envisagée. Il est trop tôt, cependant, pour la caractériser avec précision.

[1] On pourrait envisager des systèmes évolués pour l'application d'autres gestes élémentaires ou *Actions Gestuelles Elémentaires* : pousser, immobiliser tirer, etc.

RESUME ET CONCLUSION

Dans cette troisième partie nous avons :

- Proposé un parcours qui permet de comprendre les étapes fondamentales de ces processus ;
- Synthétisé les *besoins* d'un utilisateur de CORDIS-ANIMA ;
- Listé et hiérarchisé les opérations que GENESIS devra permettre et soutenir.

Nous sommes à même, maintenant, de nous consacrer plus directement à la conception de GENESIS.

Dans la partie IV, nous exposons les choix fondamentaux que nous avons opérés pour GENESIS. Nous verrons que certains de ces choix impliquent une *restriction* importante par rapport aux possibilités théoriques évoquées dans cette partie. Nous montrerons cependant qu'ils étaient nécessaires à la poursuite de nos travaux.

Les partie V et VI sont ensuite consacrées aux fonctionnalités soutenant la phase de lutherie de l'utilisateur.

La partie VII, enfin, présente les fonctionnalités de simulation et leur ergonomie.

Partie IV

Choix généraux pour l'environnement GENESIS

C’est une chose que de disposer d’un système et d’envisager tout son potentiel avec un objectif d’exhaustivité, mais c’en est une autre que de concevoir un logiciel utile, utilisable et dont les fonctionnalités puissent être apprises. C'est ainsi que de la version optimale de GENESIS évoquée dans la partie précédente il nous a fallu extraire un *sous ensemble optimal* et, pour cela, opérer avec soin quelques restrictions.

Cette partie présente les choix fondamentaux qui sont à la base de la version 1.5 de GENESIS ainsi que les considérations qui les ont motivés.

Le premier chapitre décrit la version de CORDIS-ANIMA retenue pour GENESIS.

Le second chapitre présente les quatre modes prévus pour l’environnement. Nous y exposons les raisons pour lesquelles nous avons porté l’essentiel de nos efforts sur le mode *lutherie.* Nous détaillons les différents types de données existant dans l’environnement et leur organisation – à la fois en termes de localisation dans l’arborescence du compte utilisateur et en termes de dépendances. Enfin, nous exposons les propriétés générales de *l’interface* du mode lutherie de GENESIS.

Introduction

Contexte et bibliographie

Partie I : le paradigme de la simulation interactive et multisensorielle d'objets physiques

Partie II : Interaction Homme-Machine

Partie III : Le processus de création avec GENESIS. Analyse des Tâches

Conception de GENESIS

Partie IV : Choix généraux pour l'environnement GENESIS

Chapitre 10 La version « GENESIS » de CORDIS-ANIMA 170

10.1 - RESTRICTION AU TEMPS DIFFÉRÉ 170
10.2 - FRÉQUENCE DE SIMULATION 171
10.3 - BASE DE PARAMÈTRES DANS GENESIS 172
10.4 - LA SPATIALITÉ DES OBJETS CORDIS-ANIMA ET LE CHOIX DE LA VERSION TOPOLOGIQUE 177
10.5 - LA VERSION TOPOLOGIQUE DE CORDIS-ANIMA UTILISÉE DANS GENESIS 183

Chapitre 11 Organisation générale de GENESIS 201

11.1 - LES DIFFÉRENTS MODES ET LES FONCTIONNALITÉS DE SIMULATION 201
11.2 - FICHIERS ET COMPTES UTILISATEURS 205
11.3 - L'INTERFACE DU MODE LUTHERIE 213

Résume et Conclusion 220

Partie V : Fonctionnalités fondamentales de l'atelier de lutherie GENESIS

Partie VI : Fonctionnalités évoluées du mode lutherie

Partie VII : Simulation et phénomènes

Conclusion et perspectives

Chapitre 10

LA VERSION « GENESIS » DE CORDIS-ANIMA

Toute mise en œuvre de CORDIS-ANIMA nécessite des choix préalables qui, ajoutés aux principes généraux du formalisme, définissent une *version* utilisable. Ce chapitre précise la version de CORDIS-ANIMA retenue pour GENESIS. En particulier :

- Il explique pourquoi les premières versions du logiciel ne font appel qu'au seul moteur temps différé.
- Il précise les choix relatifs à la fréquence d'échantillonnage des simulations, puis à la base de paramètres.
- En s'appuyant sur une rapide analyse de la spatialité des modèles, il montre pourquoi nous avons retenu la version topologique (ou monodimensionnelle) de CORDIS-ANIMA.
- Enfin, il présente les modules élémentaires retenus, et notamment les modules d'interaction non-linéaires, levant définitivement toute ambiguïté quant à la base formelle qui est au cœur de GENESIS.

10.1 - Restriction au temps différé

La situation d'interaction multisensorielle telle qu'elle est envisagée au sein du laboratoire offre de nombreuses possibilités dont nous avons déjà largement discuté. Sa mise en œuvre dans l'environnement GENESIS est particulièrement attendue, tant par les chercheurs que par les créateurs.

La partie V a cependant montré que des processus de création complets (i.e. : aboutissant à la création d'une œuvre sur support finalisée) tels la composition d'objets physiques ou le jeu à l'aide de déclencheurs, sont possibles avec le seul moteur temps différé. En d'autres termes, *le temps différé est suffisant pour faire de GENESIS un environnement opératoire pour la création.* Qui plus est, la grande majorité des tâches que nous avons identifiées ne dépendent pas du moteur de simulation utilisé de sorte que *la conception de la plupart des fonctionnalités de GENESIS peut être menée alors même que l'environnement est restreint au temps différé.*

Au début de nos travaux, le moteur temps réel TELLURIS[1] et un nouveau TGR© étaient encore en chantier. Leur intégration dans GENESIS aurait nécessité des efforts conséquents. Mais *la conception du système temps réel ne relève pas directement de notre problématique* ; elle constitue un projet à part entière et de longue haleine, qui a ses exigences et sa dynamique d'évolution propres. De plus, même finalisé, *le système temps réel ne serait aujourd'hui guère opératoire en situation de création.* C'est que la puissance des machines limite encore considérablement la complexité des objets qui peuvent être simulés en temps réel et donc l'intérêt *pratique* de la situation d'interaction multisensorielle.

[1] Voir [Giraud.99], [Florens&al.98].

Il est certain que temps réel et temps différé sont *complémentaires* et qu'au sein d'un environnement optimal ils devraient cohabiter. Compte tenu des considérations précédentes et en accord avec la *démarche incrémentale* retenue pour la conception de GENESIS, nous avons décidé de fonder les premières versions du logiciel exclusivement sur le moteur temps différé CORDIS-OFF.

Toutefois, précisons ici que la connexion du moteur temps-réel TELLURIS a été prévue dans l'architecture de GENESIS. Une version expérimentale de GENESIS fonctionne déjà avec ce moteur. Dans un but de clarté, la suite du document ne fera état que de l'implémentation temps différé de GENESIS.

10.2 - Fréquence de simulation

10.2.1 - Une fréquence unique : 44100 Hz.

Il serait envisageable de laisser l'utilisateur décider de la fréquence d'échantillonnage des simulations – ou des fréquences dans l'hypothèse de simulations multifréquences. Toutefois, nous avons opté dans GENESIS pour une fréquence *unique*, *non modifiable* par l'utilisateur et fixée à *44100 Hz*.

Nous considérons en effet que le choix de la fréquence de simulation constituerait pour l'utilisateur une tâche technique, qui le détournerait de ses objectifs principaux et nuirait au développement d'une *pensée physique* en mettant l'accent sur les processus numériques qui la supportent. Qui plus est, si un choix entre plusieurs fréquences était possible, un même modèle génèrerait des phénomènes différents suivant la fréquence adoptée (fondamentale, répartition des modes[1], effets non-linéaires, etc.). Cela s'opposerait indubitablement à ce qu'un *modèle mental* efficace puisse se construire relativement à CORDIS-ANIMA. Une fréquence unique évite que l'utilisateur n'ait à s'interroger sur les conséquences diverses et complexes de ses décisions en la matière.

La valeur retenue pour cette fréquence découle alors de considérations de nature technique. Il est nécessaire de choisir une fréquence de simulation élevée pour que des phénomènes haute-fréquence puissent apparaître dans les modèles (théorème de Shannon)[2]. A l'inverse, dans la mesure ou la fréquence de simulation a une influence directe sur les temps de calcul[3], il convient d'opter pour une fréquence suffisamment basse. La maîtrise des temps de calcul est en effet d'une importance cruciale dans GENESIS. Elle conditionne par exemple la possibilité d'une *démarche itérative*[4] poussée.

Le compromis que nous avons opéré s'explique en deux temps. Nous avons tout d'abord décidé de nous baser sur les standards des sons numériques de qualité musicale – c'est à dire essentiellement 44100 Hz, 48000 Hz.. Ces fréquences sont en effet adaptées à la simulation des « modèles musicaux » pour lesquels la fréquence du mode le plus bas est *en général* inférieure à 5000 Hz. Nous avons ensuite opté pour une fréquence de 44100 Hz, gagnant ainsi environ 8% de temps de calcul par rapport à une fréquence de 48000 Hz.

---

[1] [Incerti.96 pp89-95].

[2] Notons que ce n'est pas la seule raison – bien qu'elle soit suffisante. Par exemple, les causes d'instabilité dans les modèles non-linéaires s'amenuisent lorsqu'on augmente la fréquence de simulation.

[3] Puisqu'elle est égale au nombre de pas à calculer pour obtenir une seconde de son.

[4] Rappelons que la démarche itérative repose sur la boucle simulation / évaluation / modification qui, itérée, permet à l'utilisateur de converger vers le résultat souhaité – ou le cas échéant de découvrir quelques nouvelles pistes.

10.2.2 - Remarques

Deux remarques peuvent être faites quant aux choix opérés.

Il faut noter, d'abord, que certains compositeurs sont habitués à utiliser un standard spécifique pour les sons audionumériques. Ils sont alors gênés de ce que GENESIS impose une fréquence de 44100 Hz aux sons générés[1]. En réponse à cette remarque, les seules solutions seraient d'augmenter la fréquence de simulation ou de laisser l'utilisateur choisir la fréquence qu'il désire pour les sons et les simulations– éventualités dont nous avons déjà discuté.

Il convient par ailleurs d'évoquer l'éventualité de faire cohabiter plusieurs fréquence lors des simulations. Simuler à 44100 Hz les *instrumentistes* ou les *chefs*, qui sont des structures basse-fréquence ou très basse fréquence (inférieure à 10 Hz en règle générale), c'est utiliser la puissance de calcul sans aucun bénéfice. Il serait judicieux que la fréquence de simulation soit adaptée aux fréquences caractéristiques de *chaque partie* de l'objet[2]. Cela, cependant, pose de nombreux et difficiles problèmes. L'un d'entre eux est que les phénomènes générés deviendraient variables en fonction de la fréquence adoptée.

Ces remarques appellent à ce que les choix opérés quant à la fréquence des simulations dans GENESIS soient à terme reconsidérés. Il y a là matière à une recherche théorique et empirique qui aborderait des questions mathématiques et techniques (par exemple celle que pose les simulations multifréquences), mais aussi les questions d'ordre ergonomique qui apparaissent dès lors que l'utilisateur se voit offrir la maîtrise des fréquences de simulation. Cette recherche, cependant, ne nous est pas apparue comme prioritaire.

10.3 - Base de paramètres dans GENESIS

10.3.1 - Paramètres algorithmiques, paramètres CORDIS, paramètres naturels

Trois bases de paramètres (algorithmiques, CORDIS, et naturels) peuvent être considérées lorsqu'on travaille avec CORDIS-ANIMA. A chacune d'elles correspond une formulation spécifique de l'algorithme de chaque module.

Paramètres algorithmiques A, B, C

La base des paramètres algorithmiques est celle des coefficients des suites de récurrence d'ordre 0, 1 ou 2 qui constituent l'essentiel des algorithmes CORDIS-ANIMA.

Dans cette base, un module MAS (inertie) est par exemple déterminé par les trois coefficients A, B, C de son algorithme :

$$\vec{X}_{n+1} = A * \vec{X}_n + B * \vec{X}_{n-1} + C * \vec{F}_n$$

où : $\vec{X}_n$ est la position du MAS au pas n ;
$\vec{F}_n$ est la somme des forces exercées par les <LIA> connecté au MAS ;
et $\vec{X}_n$ et $\vec{F}_n$ sont des vecteurs à 1, 2 ou 3 composante(s) suivant la dimension de l'espace de simulation.

[1] Nous renvoyons, à ce sujet, aux remarques du compositeur Ludger Brumer rapportées dans l'annexe D.

[2] Notons que plusieurs recherches ont déjà abordé ce problème. Le simulateur temps réel TELLURIS est, par construction, « orienté multifréquence ».

On comprendra que la base des paramètres algorithmiques est inadaptée à GENESIS puisque les paramètres A, B, C n'offrent pas d'interprétation physique directe. Nous l'évoquons ici d'une part parce qu'elle fut au cœur des premières mises en œuvres de CORDIS-ANIMA, d'autre part parce qu'elle est encore utilisée par certaines fonctions d'allocation du simulateur CORDIS-OFF.

Paramètres CORDIS M, K, Z

La base CORDIS est celle dans laquelle la formulation des algorithmes fait abstraction de la fréquence d'échantillonnage tout en permettant un contrôle des comportements inertiels, élastiques et visqueux et une formulation simple des algorithmes. Lorsqu'on l'utilise, on fixe les *fréquences réduites* et les *temps d'amortissements réduits* des objets (c'est-à-dire mesurés relativement au pas d'échantillonnage, et non à la seconde). Donnons, pour préciser, les exemples des modules MAS et REF.

- Dans la base CORDIS, le module MAS est déterminé par un paramètre M et son algorithme est conforme à la relation :

$$M*\left(\vec{X}_{n+1}-2.\vec{X}_n+\vec{X}_{n-1}\right)=\vec{F}_n$$

où comme précédemment :

X_n est la position du MAS au pas n ;

F_n est la somme des forces exercées par les <LIA> connectés au MAS ;

et X_n et F_n sont des vecteurs à 1, 2 ou 3 composante(s).

Le paramètre M contrôle, effectivement, le comportement inertiel du MAS. Pour autant, la dimension de M n'est pas celle d'une inertie mais :

$$[M]=\frac{[inertie]}{[temps]^2}$$

- Le module REF (viscoélasticité) est déterminé par trois paramètres K, L_0, Z et répond à l'équation :

$$\vec{F}_{n,1}=\left[K*\left(D_n-L_0\right)+Z*\left(D_n-D_{n-1}\right)\right].\vec{u}_n$$

où : $D_n=\left\|\vec{X}_{n,2}-\vec{X}_{n,1}\right\|$ est l'élongation de la liaison, (distance séparant les deux points M) ;

D_n-D_{n-1} est la vitesse d'élongation de la liaison en distance/échantillon ;

$\vec{u}_n$ est le vecteur unitaire entre les deux points M connectés par le REF ;

$\vec{F}_{n,1}$ est la force exercée par les RES sur le premier point M au pas n ;

($F_{n,1}$ est le cas échéant un vecteur et D_n est un scalaire).

En modifiant K, Z ou L_0, on modifie bien les comportements élastiques ou visqueux apparents. Par contre, la dimension de Z n'est pas celle d'une viscosité : Z exprime la viscosité par pas d'échantillonnage :

$$[Z]=\frac{[viscosité]}{[temps]}$$

Paramètres naturels m, k, z

La troisième base est celle qui apparaît naturellement lorsqu'on envisage les algorithmes CORDIS-ANIMA comme les résultats de la discrétisation temporelle d'équations à temps continu (accélération centrée et vitesse retardée). Donnons, pour préciser, les exemples des modules MAS et FRO.

- Module MAS

L'équation de la masse ponctuelle à temps continu est :

$$m*\frac{d^2\vec{x}}{dt^2}=\sum \vec{f}$$

En discrétisant cette équation, on aboutit à la relation définissant le module MAS dans la base des paramètres naturels :

$$mF_e^2*\left(\vec{X}_{n+1}-2\vec{X}_n+\vec{X}_{n-1}\right)=\vec{F}_n$$

où : $\vec{X}_n$ est la position du MAS au pas n ;
$\vec{F}_n$ est la somme des forces exercées par les <LIA> connecté au MAS ;
$\vec{X}_n$ et $\vec{F}_n$ sont des vecteurs à 1, 2 ou 3 composante(s)
et : F_e est la fréquence d'échantillonnage des simulations.

- Module FRO

L'équation de la liaison visqueuse à temps continu s'écrit :

$$\vec{f}_{-1}=z*\left(d\left\|\vec{x}_2-\vec{x}_1\right\|/dt\right)\vec{u}_{1->2}$$

D'où, après discrétisation, la relation définissant le module FRO paramétré par son paramètre naturel z :

$$\vec{F}_{n,-1}=z*F_e*\left(D_n-D_{n-1}\right)*\vec{u}_n$$

où : $D_n=\text{norme}\left(X_{n,2}-X_{n,1}\right)$ est l'élongation de la liaison ;
F_e est la fréquence d'échantillonnage des simulations ;
$F_e*\left(D_n-D_{n-1}\right)$ est la vitesse d'élongation de la liaison en distance / seconde ;
$\vec{u}_n$ est le vecteur unitaire entre les deux points M connectés par le REF ;
$\vec{F}_{n,-1}$ est la force exercée par les RES sur le premier point M au pas n ;
($F_{n,-1}$ est le cas échéant un vecteur et D_n est un scalaire).

Dans ces équations, les dimensions de m, k et z sont bien celles d'une inertie, d'une raideur et d'une viscosité.

Changement de base

Les paramètres de tout modèle CORDIS-ANIMA peuvent être indifféremment exprimés dans chacune de ces bases. Le tableau de la figure 1 page suivante résume les formules de changement de base. Il donne en outre, en résumé, les algorithmes dans les trois bases de paramètres des modules CEL (oscillateur mécanique élémentaire) et REF dans le cas où *l'espace de simulation est monodimensionnel*, ainsi que la fréquence propre du CEL libre.

Base	PARAMETRES CORDIS	PARAMETRES NATURELS	PARAMETRES ALGORITHMIQUES
Changement de base depuis les paramètres CORDIS[1]	M, K, Z K_{lia}, Z_{lia}	$m=\frac{M}{F_e^2}$; $k=K$; $z=\frac{Z}{F_e}$ $k_{lia}=K_{lia}$, $z_{lia}=\frac{Z_{lia}}{F_e}$	$A=2-\frac{K+Z}{M}$; $B=\frac{Z}{M}-1$; $C=\frac{1}{M}$ $D=K_{lia}$ $E=Z_{lia}$
Changement de base depuis les paramètres NATURELS	$M=m*F_e^2$; $K=k$; $Z=z*F_e$ $K_{lia}=k_{lia}$; $Z_{lia}=z_{lia}*F_e$	m, k, z k_{lia}, z_{lia}	$A=2-\frac{k}{m.F_e^2}+\frac{z}{m.F_e}$; $B=\frac{z}{m.F_e}-1$; $C=\frac{1}{m}$; $D=\frac{k_{lia}}{F_e^2}$; $E=\frac{z_{lia}}{F_e}$
Algorithme de l'oscillateur mécanique élémentaire **CEL** soumis à la force F_n .	Paramètres : M, K, Z $X_{n+1}=\left(2-\frac{K+Z}{M}\right)X_n+\left(\frac{Z}{M}-1\right)X_{n-1}+\frac{1}{M}F_n$	Paramètres m, k, z $X_{n+1}=\left(2-\frac{k}{mF_e^2}-\frac{z}{mF_e}\right)X_n$ $+\left(\frac{z}{mF_e}-1\right)X_{n-1}+\frac{1}{mF_e^2}F_n$	Paramètres A, B, C $X_{n+1}=A.X_n+B.X_{n-1}+C.F_n$
Fréquence propre (en Hz) du module CEL en oscillation libre, ($F_n=0$)[2]	$f_p=\frac{F_e}{2\pi}*\text{Arccos}\left(\frac{2-\frac{K+Z}{M}}{2\sqrt{1-\frac{Z}{M}}}\right)\approx\frac{F_e}{2\pi}\sqrt{\frac{K}{M}}$ (Voir la note [3])	$f_p=\frac{F_e}{2\pi}*\text{Arccos}\left(\frac{2-\frac{k}{m.F_e^2}-\frac{z}{m.F_e}}{2\sqrt{1-\frac{z}{m.F_e}}}\right)\approx\frac{1}{2\pi}\sqrt{\frac{k}{m}}$ (Voir la note [4])	$f_p=\frac{F_e}{2\pi}.ArcCos\left(-\frac{A}{2\sqrt{B}}\right)$
Algorithme de la liaison viscoélastique **REF**	Paramètres K_{lia}, Z_{lia} $F_{n,2-1}=K_{lia}.D_n+Z_{lia}.(D_n-D_{n-1})$ Avec : $D_n=X_{n,2}-X_{n,1}$	Paramètres k_{lia}, z_{lia} $F_{n,2-1}=k_{lia}.D_n+z_{lia}.F_e.(D_n-D_{n-1})$ Avec : $D_n=X_{n,2}-X_{n,1}$	Paramètres D, E $F_{n,2-1}=D.D_n+E(D_n-D_{n-1})$ Avec : $D_n=X_{n,2}-X_{n,1}$

Figure 1 : changement de base de paramètres et algorithmes des modules CEL et REF topologiques (ou monodimensionnels)

[1] Ces formules sont définies à une constante prêt. On peut, bien sûr, poser m=M – les autres relations se déduisant par proportionnalité.

[2] Dans l'hypothèse où le système est oscillant.

[3] L'approximation $f_p\approx\frac{F_e}{2\pi}\sqrt{\frac{K}{M}}$ approximation est valide dans le cas où Z<<M et K>>Z.

[4] L'approximation est valide dans le cas où z<<m et k>>z. Notons que f_p (cas discret) est toujours supérieure à $\frac{1}{2\pi}\sqrt{\frac{k}{m}}$ (cas temps continu).

10.3.2 - Choix retenus pour GENESIS

Nous avons retenu la base des paramètres CORDIS pour GENESIS. Ce choix s'appuie sur plusieurs considérations :

- La base CORDIS offre une formulation particulièrement simple des algorithmes des principaux modules et permet donc de les présenter efficacement.
- Dans cette base, les paramètres nominaux permettant l'obtention de phénomènes à fréquence audio sont particulièrement simples. Par exemple, en choisissant :

  K≈ 1/10 ou 1/100 de M et Z≈ 1/10 ou 1/100 de K

  l'utilisateur est assuré d'obtenir une structure vibrante (au sens de la hiérarchie des structures)[1]. La base des paramètres CORDIS facilite ainsi la mise en œuvre de procédés mnémotechniques et l'apprentissage des valeurs nominales des paramètres.
- Enfin, les versions antérieures de GENESIS (1.0 à 1.2) reposaient sur cette base de paramètres, de sorte que les utilisateurs de GENESIS y étaient déjà habitués.

10.3.3 - Remarques

Le choix de la base CORDIS appelle deux remarques.

Tout d'abord, dans le cas de simulations multifréquences (ce que nous avons retenu comme une éventualité pour les années à venir) ou si la fréquence de simulation pouvait être modifiée, la base CORDIS deviendrait inadéquate. Les paramètres CORDIS déterminent les constantes de temps réduites, de sorte que les constantes de temps exprimées en Hertz ou en seconde pour un ensemble donné de paramètres sont *proportionnelles* à la fréquence de simulation. Dès lors, deux objets identiques (i.e : ayant la même structure et les mêmes paramètres CORDIS) mais simulés à des fréquences différentes génèrent des phénomènes différents. Or, il est certain que l'expertise de l'utilisateur en matière de contrôle des paramètres doit pouvoir se développer relativement aux phénomènes générés. Cela nécessite, précisément, qu'à certaines plages de paramètres corresponde *toujours* certains types de comportement, par exemple ceux qui caractérisent une structure vibrante, un chef, etc. La base CORDIS serait donc à exclure si plusieurs fréquences de simulations cohabitaient dans GENESIS[2].

La deuxième remarque est relative à la mesure des vitesses. Le choix de la base CORDIS implique que dans l'algorithme du FRO la vitesse d'élongation de la liaison soit mesurée en unité de distance parcourue par pas d'échantillonnage. C'est cette unité qui doit être choisie en abscisse pour représenter la *courbe caractéristique* du module FRO, c'est à dire la force exercée en fonction de la différence de vitesses des deux points M connectés (figure 2).

---

[1] Ceci pour une fréquence d'échantillonnage fixée à 44100 Hz.

Pour l'oscillateur mécanique élémentaire en oscillation libre, par exemple, les valeurs :

M=1 ; K=0.01 ; Z=0.0001

Induisent une *fréquence réduite* Fr≈0.016 donc une fréquence fondamentale d'environ 0.016*44100 = 702 Hz et un temps d'amortissement de 0,45 s.

[2] Précisons cependant que le choix de la base naturelle n'apporterait qu'une réponse partielle au problème que poserait la cohabitation de plusieurs fréquences. Dans cette base, en effet, des simulations à des fréquences différentes de deux modèles identiques (i.e. : ayant la même structure et les mêmes paramètres naturels) ne donneraient des phénomènes comparables *qu'en première approximation*.

Prenons l'exemple de l'oscillateur mécanique élémentaire CEL. Pour des paramètres naturels :

m=1 ; z =1000 ; k=1^{c}8

la fréquence propre est F_{44100} ≈ 1600 Hz pour une fréquence de simulation Fs=44100 Hz

et F_{48000} ≈ 1602 Hz pour une fréquence de simulation Fs=48000 Hz

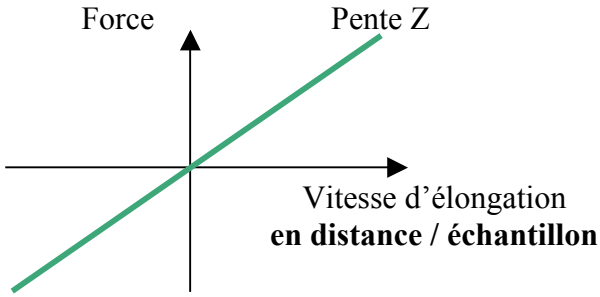


Figure 2 : représentation de la courbe caractéristique du module FRO dans la base de paramètres CORDIS

Pourtant, il est bien certain que pour l'utilisateur une vitesse se mesure relativement à la seconde et non relativement au pas d'échantillonnage. Dès lors que la courbe caractéristique lui est accessible, comme dans le cas des modules non-linéaire LNL que nous introduisons plus bas, la base des paramètres CORDIS devient problématique : elle impose la cohabitation de deux unités de vitesse au sein de l'environnement.

10.4 - La spatialité des objets CORDIS-ANIMA et le choix de la version topologique

Un autre choix fondamental lorsqu'on utilise CORDIS-ANIMA est celui de la spatialité des objets.

10.4.1 - Espace des mouvements, géométrie, espace de simulation : les trois caractères de la spatialité

Un objet CORDIS-ANIMA évolue dans un *espace de simulation*, somme de son *espace des mouvements* et de son *extension spatiale*[1]. L'espace des mouvements est l'espace dans lequel les points M de l'objet se *déplacent* ; il conditionne les degrés de liberté de chaque point M. L'extension spatiale est l'espace que l'objet occupe au repos c'est-à-dire sa géométrie. L'espace de simulation, somme des deux précédents, est l'espace qu'il faut considérer pour simuler l'objet. Toutes les combinaisons de ces différents espaces sont permises par le formalisme.

Le cas le plus général est celui des objets tridimensionnels qui occupent au repos les trois dimensions et dont les points M ont trois degrés de liberté. On peut cependant définir des objets bidimensionnels, ou encore décider que l'objet occupe au repos deux dimensions de l'espace, mais que ses points M se déplacent suivant l'axe perpendiculaire au plan de l'extension spatiale (figure 3). Dans ce dernier cas l'espace de simulation est à nouveau tridimensionnel, mais le calcul d'un pas de simulation ne nécessite une mise à jour de la position des points M que suivant le seul axe des mouvements.

[1] Voir [Castagne.96], [Castagne&Cadoz.98].

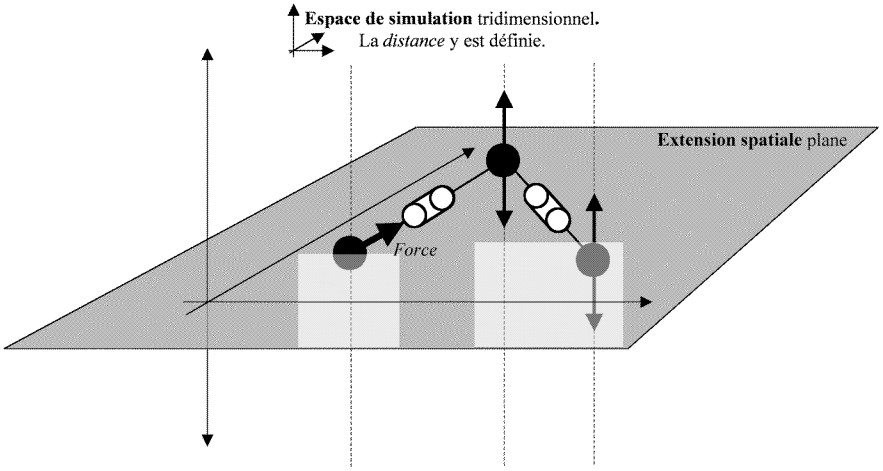


Figure 3 :un cas de spatialité complexe.
Ici, l'espace des mouvements monodimensionnel est perpendiculaire à l'extension spatiale plane et l'espace de simulation est tridimensionnel.

Un cas très particulier de spatialité est celui où l'espace de simulation est monodimensionnel (figure 4). Dans ce cas, quelle que soit la complexité du réseau, une seule dimension X intervient dans les calculs. Au repos, tous les modules sont en général regroupés en une seule et même position de l'espace. Définir un objet, c'est donc essentiellement définir le *réseau* <MAT>-<LIA>. C'est la raison pour laquelle on parle alors *d'objet topologique* et de la *version topologique* de CORDIS-ANIMA.

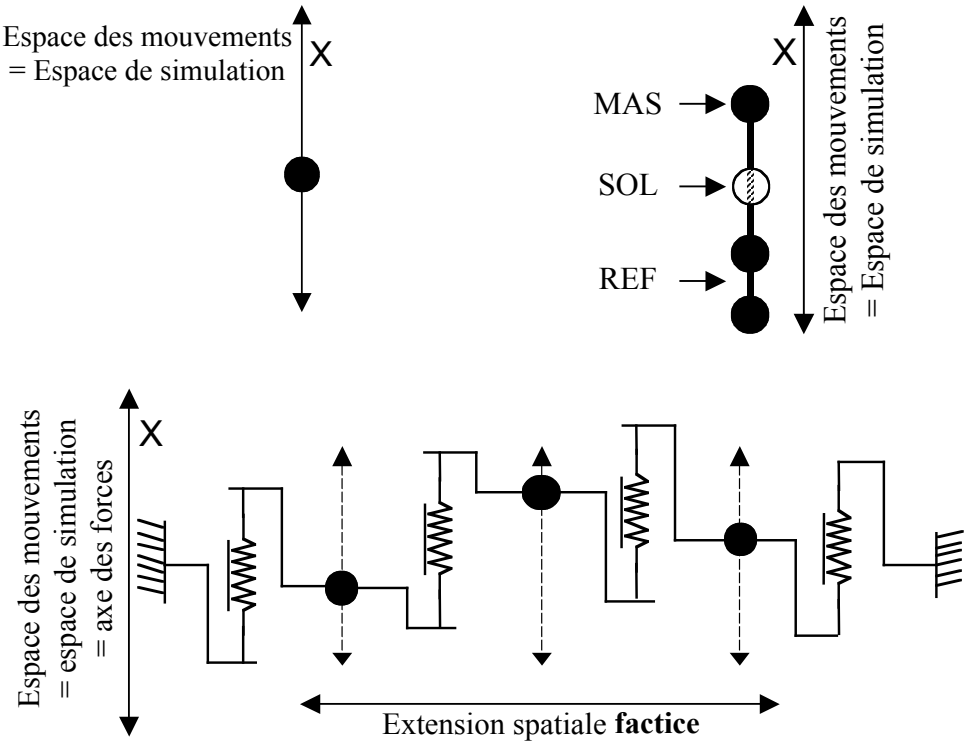


Figure 4 : deux représentations d'un objet topologique
en haut : dans l'espace de simulation monodimensionnel,
au repos (tous les <MAT> sont concentrés au même point de l'espace) puis en mouvement
en bas : représentation fonctionnelle avec une extension spatiale factice.

10.4.2 - Quelques enjeux de la spatialité.

Comment choisir ces différents espaces lorsqu'on souhaite mettre en œuvre CORDIS-ANIMA ? L'annexe 1 du document propose l'étude d'un filament bidimensionnel, et met ainsi en évidence par l'exemple l'intérêt des objets *spatiaux* (dont la dimension de l'espace de simulation est supérieure à 1) pour la création sonore et musicale. Le lecteur pourra s'y reporter pour de plus amples précisions et nous ne retenons ici que ses principales conclusions.

Spatialité et complexité algorithmique[1]

La spatialité a un impact de toute première importance sur la complexité des calculs ; elle conditionne d'une part la taille de l'état du modèle, d'autre part la complexité du calcul de la distance.

L'état d'un modèle à un moment donné est défini par l'ensemble des positions et positions retardées de chacun de ses points M suivant les axes de l'espace des mouvements. Le nombre des variables d'état à mettre à jour à chaque pas de simulation est proportionnel à la dimension de l'espace des mouvements.

Le *calcul de distance*, nécessaire à la détermination des forces exercées par les <LIA>, est une opération très courante lors des simulations. Sa complexité est déterminée par la dimension de l'espace de simulation. Dès lors que cette dimension est supérieure à 1, l'évaluation d'une distance fait appel à la fonction racine carrée, dont le coût est élevé. Au contraire, lorsque l'espace de simulation est monodimensionnel, c'est à dire dans le cas des objets topologiques, le calcul de distance est extrêmement simplifié : il se résume au calcul de la différence des deux positions considérées sur l'axe des mouvements (ou axe de simulation).

En conclusion, nous retiendrons que le choix de la version topologique de CORDIS-ANIMA permet, pour une complexité donnée du réseau <MAT>-<LIA>, un gain considérable sur la complexité des calculs.

Spatialité, complexité formelle, représentation mentale

Il existe une relation non triviale entre la spatialité des objets et la complexité formelle qui en résulte et donc entre la spatialité et la difficulté de mise en œuvre du processus de modélisation.

Le cas le plus aisé à appréhender est probablement celui où chacun des espaces (extension spatiale, espace des mouvements et donc espace de simulation) est tridimensionnel. C'est lui, en effet, qui s'approche au mieux de l'espace « réel » dans lequel nous évoluons.

Le cas ou l'extension spatiale de l'objet et l'espace des mouvements sont orthogonaux (figure 3 page précédente) est au contraire particulièrement peu intuitif.

Le cas de la version topologique est encore plus particulier. Construire un objet topologique suppose de réfléchir essentiellement en termes de réseau et de connexion, et non pas en termes de géométrie. Il est certain que le chemin que doit accomplir un utilisateur pour « penser correctement » les modèles topologiques n'est pas évident. La

[1] Lorsque la complexité algorithmique de deux modèles sonores est comparée dans ce paragraphe, nous supposons qu'ils ont tous deux le même de nombre de <MAT> et de <LIA>. C'est là une hypothèse valide : dans un modèle CORDIS-ANIMA, *le nombre de modes* d'une structure est déterminé par le nombre de ses <MAT> mobiles, de sorte que deux modèles constitués du même nombre de modules auront, *en première approximation*, une richesse sonore comparable, et ce indépendamment de leur spatialité.

modélisation topologique s'avère cependant parfaitement praticable dès lors que l'utilisateur en a compris le principe.

De façon plus pragmatique, il faut remarquer que le nombre de paramètres à spécifier (et donc auxquels il faut réfléchir) augmente avec la complexité de la spatialité.

Dans les modèles spatiaux, il faut pour définir un objet non seulement spécifier le réseau <MAT>-<LIA>, les modules et leurs paramètres, mais encore préciser la géométrie (ou extension spatiale) de l'objet en saisissant les *positions initiales* de chacun des <MAT> suivant chacune des dimensions de l'espace. Par ailleurs, un paramètre de *longueur au repos* des liaisons, notamment des liaisons élastiques est nécessaire.

Ce dernier paramètre disparaît dans le cas de la version topologique. Les conditions initiales d'un point M sont alors réduites à la position initiale et la vitesse initiale suivant l'axe des mouvements. En général, tous les <MAT> du modèle sont concentrés au repos en un seul point de l'espace de simulation. Alors que dans le cas spatial les conditions initiales participent de la *nature* de l'objet (de ce qu'il *est*), elles ne servent plus guère dans le cas topologique à spécifier la géométrie de l'objet mais uniquement les conditions de son *excitation*. Elles ne font plus directement partie de l'acte de modélisation lui-même et peuvent être déterminées séparément.

En conclusion de ce paragraphe, nous retiendrons :

- Que la spatialité la plus facile à appréhender est le cas ou l'objet est tridimensionnel ;
- Que la version topologique de CORDIS-ANIMA restreint le nombre des paramètres que doit régler l'utilisateur pour concevoir l'objet. Elle permet une utilisation des conditions initiales tournée vers le travail de l'excitation et donc vers un « jeu » élémentaire.
- Qu'un modèle mental efficace et opératoire quoi qu'un peu difficile à appréhender est possible face à la version topologique. Dès lors qu'il est intégré par l'utilisateur, celle-ci devient même particulièrement aisée à mettre en œuvre puisque, avec elle, toute notion d'espace s'efface pratiquement de l'acte de modélisation.

<u>Spatialité, contact entre objets, non-pénétrabilité</u>

Considérons un instant le modèle 2D de la figure 5, dans lequel on souhaite percuter un objet : un <MAT> (1) est lancé sur un <MAT> (2), (1) et (2) étant reliés par un module BUT[1].

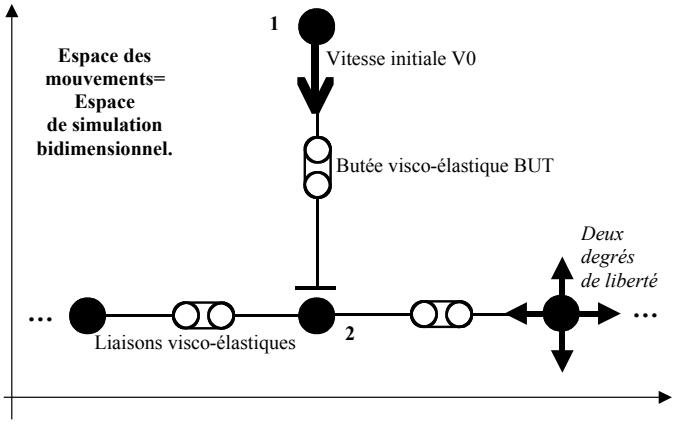
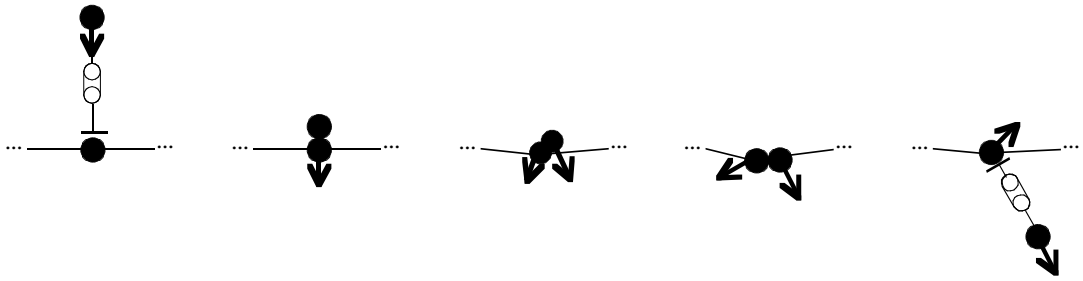


Figure 5 : modèle bidimensionnel pour une percussion élémentaire.

[1] Rappelons que le module BUT (ou *butée visco-élastique*) permet de modéliser le contact entre objets en conditionnant l'établissement d'une interaction viscoélastique à la distance des deux points M connectés (voir la partie I ou le paragraphe 10.5.2 - ci dessous)

L'histogramme de la figure 6 présente un déroulement possible de la phase d'interaction.



a) incidence | b) début de l'interaction | c) phase d'interaction visco-élastique. | d) le percuteur excite également l'axe longitudinal | e) fin de l'interaction. Le percuteur a traverséla structure

Figure 6 : déroulement d'une percussion d'un modèle 2D.
dans le cas représenté ici, le percuteur traverse la structure vibrante du fait de l'excitation longitudinale du modèle.

Il apparaît que la propriété de non-pénétrabilité de deux objets n'est pas aisée à obtenir dans les modèles masse-interaction lorsque l'espace des mouvements est pluri-dimensionnel. Plusieurs solutions permettent bien sûr de contourner ce problème – on peut, notamment, choisir une grande valeur pour le seuil S du module BUT et ainsi « entourer » le percuteur par une sphère d'interaction de taille importante. On comprendra cependant que la réalisation d'une percussion ou encore la détermination d'une *surface non pénétrable* autour d'un objet est grandement facilitée lorsque l'espace des mouvements est monodimensionnel. C'est le cas, notamment, avec la version topologique de GENESIS.

Spatialité, non-linéarité, modes de vibration

Nous envisageons ici quelques-unes des propriétés induites par la spatialité sur les phénomènes (notamment sonores) générés lors des simulations.

Dans le cas des objets topologiques, le calcul de la distance est *linéaire*. En conséquence, un objet topologique construit avec des modules linéaires (MAS, SOL, RES, FRO…) est lui aussi linéaire : sa réponse est directement proportionnelle à l'intensité de l'excitation qui lui est appliquée[1]. Deux excitations de même nature mais d'énergies différentes produiront des phénomènes identiques à un facteur d'amplitude près.

Ce n'est plus le cas dès lors que l'espace de simulation est bi ou tri-dimensionnel et que le calcul de distance, qui fait alors intervenir la fonction racine carrée, est non linéaire[2]. Le timbre des sons générés est alors corrélé au niveau d'excitation. Qui plus est, lorsque l'espace des mouvements est lui-même pluri-dimensionnel, plusieurs modes de vibration cohabitent[3] dans l'objet, et un couplage des différents modes apparaît.

[1] Ce point est démontre au chapitre II de la partie VIII, qui montre que l'algorithme complet d'un objet topologique peut être écrit sous la forme d'un système matriciel linéaire.

[2] L'Annexe B met en évidence ces diverses propriétés et leur intérêt pour la création sonore et musicale à travers l'étude d'un cas particulier (copie de la référence [Castagne&Cadoz.00]).

[3] Si la géométrie de l'objet est simple, par exemple plane ou linéique, on peut alors parler de modes longitudinaux et transversaux.

Les comportements non linéaires induits par les spatialités non triviales sont très intéressants du point de vue perceptif : ils permettent une plausibilité et une richesse accrue des sons. Il convient cependant de remarquer que l'introduction raisonnée de liaisons non linéaires dans les modèles topologiques permet de retrouver à moindre coût certaines de ces propriétés[1]. Mais ces non-linéarités devront alors être explicitement modélisées alors qu'elles apparaissent spontanément dans les modèles spatiaux.

Spatialité, déformation, geste

Nous évoquerons pour terminer la relation existant entre la spatialité des modèles et leur aptitude à être déformés par un geste de modification.

Dans les modèles spatiaux, l'application d'une tension de précontrainte suivant l'un des axes de l'extension spatiale de l'objet modifie les propriétés vibratoires des modes orthogonaux. Par exemple, en « tirant » sur un modèle 2D de corde (i.e. : un chapelet 2D), on module la fondamentale des modes transversaux[2]. Le contrôle de la déformation (ou de la précontrainte) par un geste est musicalement pertinent. Que ce geste soit exercé en temps réel avec interaction ou en temps différé, par un modèle d'instrumentiste par exemple, il constitue un processus physique à la fois facile à appréhender et permettant de moduler les propriétés sonores des structures de façon efficace et pertinente.

Un tel geste de modification n'est pas possible sur un modèle topologique. Lorsqu'on déforme un modèle topologique constitué de modules linéaires suivant l'axe de simulation, les forces de raideur exercées par les <LIA> du modèle acquièrent certes une composante continue, mais les modes de propagation ne sont en rien modifiés, pas plus que les propriétés sonores.

En conclusion, nous retiendrons que les spatialités non triviales offrent d'importantes possibilités en matière de contrôle gestuel. A l'inverse la version topologique de CORDIS-ANIMA restreint les accès gestuels et donc la variabilité des sons.

10.4.3 - Expériences ; décisions

Lorsqu'on s'apprête à mettre en œuvre CORDIS-ANIMA, le choix de la spatialité des objets adresse donc plusieurs enjeux. Les modèles spatiaux présentent des caractéristiques intéressantes, notamment en termes de non linéarité et de contrôle gestuel. Mais la version topologique offre aussi des avantages, qu'on peut résumer en deux points :

- Elle apporte, toute choses égales par ailleurs, un gain considérable en termes de complexité algorithmique ;
- Après une phase d'adaptation passée, elle s'avère particulièrement aisée à mettre en œuvre.

Ces deux points sont essentiels pour notre propos. Le premier conditionne la possibilité d'une démarche itérative poussée, et le second la facilité de prise en main du formalisme.

Par ailleurs, la plupart des recherches et expérimentations qui ont été pratiquées au sein du laboratoire en matière de création sonore et musicale sont basées sur la version

[1] Voir à nouveau, par exemple, l'Annexe B.

[2] Une approche physicienne permet de dire qu'on module la « raideur » des modes transversaux, ou encore la « raideur » des liaisons du modèle topologique *équivalent* au système des ondes transversal. Voir l'Annexe B du document.

topologique de CORDIS-ANIMA. Elles ont prouvé que la diversité et la qualité des sons et plus généralement des phénomènes qui peuvent être engendrés par les modèles topologiques sont tout à fait satisfaisantes. Elles ont permis en outre de développer une connaissance théorique et un savoir-faire relativement à la modélisation topologique dont nous pouvons largement bénéficier pour concevoir GENESIS. Suite à ces expériences, une volonté forte de disposer d'outils plus performants pour créer avec la version topologique s'est fait jour.

Compte tenu des arguments développés ici et bien que considérant que GENESIS devrait permettre, à terme, de mettre en œuvre indifféremment tous les cas de spatialité[1], *nous avons décidé de fonder pour l'heure GENESIS sur la version topologique de CORDIS-ANIMA.*

10.5 - La version topologique de CORDIS-ANIMA utilisée dans GENESIS

Pour que la version de CORDIS-ANIMA utilisée par GENESIS soit intégralement spécifiée, il nous faut encore préciser les modules que nous avons retenus. Nous aborderons successivement dans ce paragraphe :

- Les modules physiques linéaires
- Les modules physiques non-linéaires
- Les modules de mesure et de sortie sonore
- Les modules d'entrée et d'interaction
- Les modules fonctionnels

Enfin, nous synthétisons les 10 modules retenus dans la version 1.5 de GENESIS et les 3 modules accessibles dans certaines versions ad hoc.

10.5.1 - A propos des modules linéaires élémentaires et intégrés

CORDIS-ANIMA définit les notions de modules élémentaires et de modules intégrés, réunion de plusieurs modules élémentaires.

Il existe quatre modules physiques linéaires élémentaires :

- Le MAS – masse mobile ;
- Le SOL – point fixe, masse d'inertie infinie ;
- Le RES – élasticité ;
- Le FRO – viscosité ;

Dans la version topologique de CORDIS-ANIMA, il existe en outre deux modules linéaires intégrés :

- Le REF = RES + FRO – viscoélasticité
- Le CEL = SOL + REF + MAS – oscillateur mécanique élémentaire autour du point fixe SOL placé à la position $X_{SOL} = 0$ de l'axe des mouvements.

[1] Nous aurons l'occasion, dans la Conclusion du document, de proposer quelques pistes en ce sens.

La coexistence des modules élémentaires et intégrés procède, à l'origine, de considérations algorithmiques : il est plus rapide de simuler un module intégré que l'ensemble des modules élémentaires qui le constituent et réciproquement de simuler un module élémentaire qu'un module intégré dans le cas où ce dernier est dégénéré (certains de ses paramètres étant nuls). Il convient donc que modules intégrés et modules élémentaires cohabitent au sein du *simulateur* de GENESIS.

Cela, cependant, n'implique pas que les modules élémentaires et intégrés soient tous accessibles à l'utilisateur : il pourrait toujours n'utiliser que des modules élémentaires en les superposant le cas échéant (et réciproquement n'utiliser que des modules intégrés éventuellement dégénérés). En décidant de ne conserver que certains modules, on réduirait le nombre de primitives accessibles à l'utilisateur et on pourrait ainsi espérer diminuer la charge cognitive induite par l'utilisation du logiciel.

Ce n'est pas l'option que nous avons retenue, pour trois raisons :

- L'apprentissage de GENESIS et de CORDIS-ANIMA nécessite que l'utilisateur puisse expérimenter les primitives physiques les plus simples et les effets de leurs paramètres, donc qu'il ait accès aux modules élémentaires.
- Il arrive qu'il soit nécessaire de recourir à un module exclusivement visqueux ou élastique. Ces cas sont assez rares. Ils reposent sur une intention explicite de l'utilisateur et méritent d'être singularisés dans GENESIS, ce qui implique que les modules élémentaires restent disponibles.
- Par contre, les liaisons viscoélastiques REF sont les plus courantes dans les modèles CORDIS-ANIMA – puisque la matière réelle est à la fois élastique et visqueuse. De même, outre qu'il est le plus petit objet oscillant (sorte de « primitive physique sonore »), le module CEL permet de modéliser à moindre coût la raideur et la viscosité de milieu ou encore certains objets simples tels les chevalets.

En conséquence, nous avons décidé de conserver dans GENESIS l'ensemble des modules élémentaires MAS, SOL, RES, FRO et des modules intégrés CEL et REF.

10.5.2 - A propos des modules non-linéaires

CORDIS-ANIMA met prudemment de côté la question des non-linéarités en définissant le formalisme très général du module LIC (Liaison Conditionnelle)[1]. Deux remarques sont ici nécessaires :

- Tout d'abord, il n'est pas évident que le formalisme du LIC soit général, c'est à dire qu'il permette de modéliser toutes les interactions non-linéaires intéressantes.
- Ensuite, le LIC est un module théorique : il n'est pas disponible dans son intégralité dans les moteurs de simulation et n'est jamais mis en œuvre tel quel. Chercheurs et étudiants ont, au gré des nécessités, implémenté au sein des moteurs de simulation CORDIS-OFF et TELLURIS des modules non-linéaires ad-hoc[2].

[1] Rappelons que LIC est défini par un automate d'états finis. A chaque état correspond une liaison particulière définie par deux caractéristiques Force/Position et Force/Vitesse – sans précision sur la façon dont ces caractéristiques sont spécifiées. La transition d'un état à l'autre est déterminée par des conditions de passage portant sur l'élongation ou la vitesse d'élongation de la liaison – sans précision sur l'expression de ces conditions.

[2] Par exemple : le module BUT, viscosités ou élasticités non-linéaires, viscosités dépendant de la position, hystérésis, etc.

Quand bien même il serait disponible dans les moteurs de simulations, nous considérons que le formalisme du module LIC serait trop complexe pour être pratiqué au sein de GENESIS par un utilisateur non-scientifique. Nous ne pouvons pas non plus nous permettre de multiplier les types de modules accessibles à l'utilisateur en proposant un grand nombre des implémentations ad hoc conçues années après années au sein du laboratoire. Il nous appartient donc de restreindre le formalisme du LIC en définissant quelques modules spécifiques à GENESIS. Deux critères sont essentiels ici :

- Un critère d'utilité ou de généralité. Ces modules doivent permettre de modéliser les interactions non-linéaires les plus utiles. Il nous faut donc nous baser sur ce qu'on peut imaginer des besoins de l'utilisateur. Cette approche étant probablement limitée, il nous faut de plus chercher à permettre un maximum de liberté et de possibilités.
- Un critère d'utilisabilité, de facilité de compréhension, de manipulation d'apprentissage, etc.

Commençons par évoquer le premier de ces critères. Nous distinguons trois enjeux majeurs pour les interactions non-linéaires dans les modèles topologiques musicaux.

Interactions entre objets

Les interactions entre objets (réels ou CORDIS-ANIMA) sont, très souvent, non-linéaires. Dans le cadre de la modélisation physique pour la synthèse musicale, on pensera par exemple aux modalités d'excitation canoniques que sont la percussion, le pincement ou encore le frottement. De façon plus générale, il nous faut permettre la modélisation des interactions non linéaires très diverses apparaissant lorsque deux objets sont en contact.

Non linéarité de la matière réelle

La matière n'est linéaire qu'en première approximation. Lorsqu'il étudie un matériau, le physicien en cherche d'abord une modélisation linéaire, mais il incorpore rapidement dans ses modèles des composantes non linéaires. Le module d'Young, par exemple, n'est constant qu'en première approximation. Au sein de CORDIS-ANIMA, les modules non-linéaires doivent permettre la modélisation de ce type de propriétés.

Rupture de la linéarité des modèles topologiques

Comme nous l'avons indiqué, un modèle topologique constitué de modules linéaires est lui-même linéaire et cette propriété tend à induire une certaine platitude dans les sons générés. Il est possible de la contourner en insérant, au cœur des objets oscillants, quelques non-linéarités bien choisies.

Une première approche, disons physicienne, consiste à chercher à modéliser dans l'univers topologique certaines propriétés propres aux objets spatiaux – par exemple la non linéarité de la raideur des modes transverses dans les structures précontraintes. Mais l'enjeu est cependant ici beaucoup plus général. L'essentiel est qu'un comportement non proportionnel au niveau d'excitation puisse être obtenu. Pour cela, l'utilisateur peut adopter une démarche empirique et exploratoire, sans plus chercher à s'appuyer sur une référence réelle. Les non-linéarités sont ainsi susceptibles de devenir dans GENESIS l'une des voies d'expression de la créativité et de l'imagination. Encore faut-il, pour que ce soit possible, que les modules proposés soient suffisamment ouverts et faciles d'emploi.

Pour définir, face à ces enjeux, les modules implémentés dans GENESIS, nous avons adopté une approche essentiellement pragmatique visant un certain équilibre entre difficulté de mise en œuvre pour l'utilisateur et généralité. Nous avons d'abord déterminé un classement des types d'interactions non-linéaires en fonction de leur importance et de leur récurrence probable dans les modèles sonores, puis cherché une base formelle réduite (i.e. : un petit ensemble de modules) permettant la mise en œuvre des principales. Cette étude nous a conduit à implémenter trois modules non-linéaires dans GENESIS : les modules BUT et LNL et la raideur non linéaire en X^3, que nous présentons maintenant.

Le contact : la butée viscoélastique BUT

Nous avons tout d'abord retenu le module BUT, déjà défini dans les textes de référence de CORDIS-ANIMA, afin de permettre l'expression du contact entre objets lorsqu'ils se rapprochent. Rappelons que le BUT conditionne l'établissement d'une interaction viscoélastique simple à la position relative des deux <MAT> connectés sur l'axe de simulation. Il est définit par trois paramètres K, Z et S (seuil) et par l'algorithme :

```
Si (X2-X1<S)
Alors liaison viscoélastique de paramètre K et Z
Sinon interaction nulle
```

Où X1 et X2 sont les positions des deux <MAT> auxquels le BUT est connecté.

Citons, en matière d'exemples, trois utilisations possibles du BUT :

- La percussion

Le BUT permet à deux structures de se percuter. Les paramètres K et Z conditionnent alors la qualité de la percussion.

- L'étouffement

En choisissant K=0, l'interaction qui s'établit lorsque les deux structures entrent en contact est uniquement visqueuse. Leur énergie est alors dissipée proportionnellement à la valeur de Z.

- Les structures maracassées

Une *structure maracassée* est obtenue en substituant dans un objet un grand nombre de <LIA> viscoélastiques par des BUT. Le comportement de l'objet d'origine est en partie conservé, mais des propriétés chaotiques s'y superposent. Le nombre de <LIA> substitués permet de contrôler l'importance du phénomène.

Raideur et viscosité non-linéaires : le module LNL

La *raideur non linéaire* (force dépendant des positions relatives des deux <MAT>) et la *viscosité non linéaire* (force dépendant des vitesses relatives des deux <MAT>) sont deux autres catégories de non-linéarité essentielles. Elles nécessitent qu'un module <LIA> particulier soit défini, dont les caractéristiques force/position et force/vitesse puissent être modifiées, de telle sorte que la force exercée totale s'exprime par :

$$F_{1} = f_r(X_2 - X_1) + f_v(V_2 - V_1)$$

où : f_r et f_v sont deux fonctions non-linéaires définies sur l'ensemble des réels.

Deux options étaient possibles pour la spécification des caractéristiques f_r et f_v :

- Une définition algébrique ;
- Une définition par des poignées ou *point par point*.

La première de ces options pose des problèmes certains d'implémentation. Elle nécessite, d'abord, qu'une gestion des cas particuliers soit mise en œuvre. Par exemple, une hyperbole n'est pas définie en un point et, en conséquence ne peut être simulée telle quelle : il faut ajouter à la fonction algébrique la valeur que doit prendre la force au point singulier. Elle suppose, ensuite, que le moteur de simulation connaisse le langage mathématique usuel, sa syntaxe et sa grammaire.

Par ailleurs, de nombreux types d'interactions ne peuvent pas être définis par une fonction unique, mais supposent au contraire que plusieurs fonctions se succèdent[1] suivant l'axe des positions ou des vitesses. L'approche algébrique ne saurait donc se limiter à la donnée d'une seule fonction sur les réels.

Enfin, il faut remarquer que l'idée d'une définition algébrique des caractéristiques est peu conforme à « l'esprit CORDIS-ANIMA ». Elle suppose une approche *fonctionnelle* de la modélisation et serait très probablement extrêmement difficile à mettre en œuvre par des utilisateurs non-scientifiques.

Nous avons donc opté pour une définition *point par point* des courbes caractéristiques, autorisant ainsi qu'elles soient éditables par manipulation directe. Par contre, une simple interpolation linéaire entre les points s'est avérée insuffisante : elle induit une discontinuité dans la dérivée des courbes caractéristiques qui, parfois, s'entend dans le résultat sonore. Il était donc nécessaire de permette un *lissage* des caractéristiques.

Le module LNL (pour Liaison Non Linéaire) que nous avons ajouté au formalisme est donc défini :

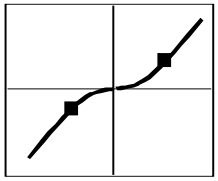
- par deux tableaux de points à deux dimensions, le premier définissant les poignées de la courbe f_r dans le plan Position/Force et le second celui de la courbe f_v dans le plan Vitesse/Force.
- par des interpolations entre chaque point des deux tableaux. Nous avons retenu trois types d'interpolations : *linéaire* (la plus simple) *par des splines*, c'est à dire des portions de fonctions polynomiales en de degré 3 (pour assurer la continuité de la dérivée), *hyperbolique* (essentiellement pour permettre que les modèles traditionnels de frottement d'archet puissent être implémentés).

La figure 7 expose en pseudo-code l'algorithme du LNL. Pour ne pas alourdir inutilement la présentation, nous ne présentons que le cas de la *raideur* non linéaire.

[1] Pour s'en convaincre, il suffit de regarder la figure 7 plus loin qui montre quelques exemples d'utilisation du module LNL retenu.

```
// les interpolations possibles entre points
Constante LINEAIRE 0 ;   SPLINE 1 ;   HYPERBOLIQUE 2

// Fonction d'allocation auprès du simulateur
// Exemple : 3 segments linéaire, spline, linéaire.
//    NbSegElas = 3    (numérotés 1, 2 et 3)
//    PtsElas =        [AbsicePt1 ; AbsicePt2]
//    InterpolElas =   [LINEAIRE ; SPLINE ; LINEAIRE]
//    CoeffElas =      [PenteDroite1 ; OrdOrigineDroite1 ; 0. ; 0. ;
//                      CoeffSpline2A ; CoeffSpline2B ; CoeffSpline2C ; 0. ;
//                      PenteDroite3 ; OrdOrigineDroite3 ;0. ;0.]
```



```
Fonction entier AllocLNL(  // paramètres de l'élasticité NL
    Entier NbSegElas                        // nombre de segments >=1
    Réel   LimitesSeg  [NbSegElas-1]        // les poignées
    Entier InterpolElas[NbSegElas]          // type d'interpolation pour chaque segment
    Entier CoeffElas   [4*NbSegElas]        // coefficients d'interpolation)
{
    // Vérifie la validité des données, s'assure notamment
       que les interpolations hyperboliques sont bien définies sur le segment considéré
    Si  // il y a une erreur Retour 0
    // Alloue le module auprès du simulateur et copie les paramètres
    Retour 1      // correctement alloué
}

// Macros utilitaires pour le calcul de la force
Macro réel Linéaire(X,A, B):      (A*X + B)
Macro réel Spline(X, A, B, C):    (A*X*X*X + B*X*X + C)
Macro réel Hyperb(X, A, B, C, D): (A*X + B) / (C*X + D)

// Fonction d'exécution du module. Elle calcule et retourne la force
Fonction réel ExecLNL(Réel DeltaX  // différence de position des MAT connectés)
{
réel ForceElas;
Statique réel NumSeg <- 1;   // variable statique pour optimisation
                             // on commence la recherche la ou on était au dernier passage
entier Trouve <- 0;
    // Repérer le segment du tableau de point PtsElas qui contient DeltaX
    TantQue (Trouve = 0)
    {
      Si (NumSeg > 1 ET NbSegElas > 1)
           Si (X < PtsElas[NumSeg].Pos)  // il faut regarder avant
                NumSeg <- NumSeg - 1;
      Sinon (NumSeg < NbSegElas)
           Si (X > PtsElas[NumSeg].Pos)
                NumSeg <- NumSeg + 1;    // il faut regarder après
      Sinon trouve = 1;
    }
    // Calculer la force en fonction de l'interpolation sur ce segment
    Selon InterpolElas[NumSeg]
    {
      CAS LINEAIRE
           retour Lineaire(    DeltaX, CoeffElas[4*(NumSeg-1) + 1,
                               CoeffElas[4*(NumSeg—1) + 2])
      CAS SPLINE
           retour Spline(      DeltaX, CoeffElas[4*(NumSeg-1) + 1,
                               CoeffElas[4*(NumSeg-1) + 2],
                               CoeffElas[4*(NumSeg-1) + 3])
      CAS HYPERB
           retour Hyperb(      DeltaX, CoeffElas[4*(NumSeg-1) + 1,
                               CoeffElas[4*(NumSeg-1) + 2],
                               CoeffElas[4*(NumSeg-1) + 3],
                               CoeffElas[4*(NumSeg-1) + 3])
    }
}
```

Figure 7 : algorithme de la partie Raideur Non Linéaire du LNL en pseudo code (les tableaux de taille N sont indexés de 1 à N).

Etant très largement paramétrable, le module LNL permet d'exprimer une grande diversité d'interactions non-linéaires. Nous terminerons ce paragraphe en évoquant quelques exemples d'utilisations canoniques (voir également la figure 8 page suivante) :

- Le REF : en choisissant comme caractéristique deux droites de pentes K et Z passant par l'origine, il est possible de reproduire un module REF de paramètres K et Z[1]. Cela peut grandement faciliter la compréhension des axes des abscisses et des ordonnées des deux caractéristiques par l'utilisateur[2].

- Viscoélasticité non-linéaire : utilisé pour rompre la linéarité des caractéristiques de viscosité et de raideur, le LNL peut être inséré au cœur des objets oscillants. Partant d'une structure initialement linéaire, il est intéressant de substituer quelques modules REF par des liaisons LNL, puis de modifier petit à petit les caractéristiques de raideur et de viscosité en évaluant à l'écoute le résultat.

- Frottement d'archet : le frottement d'archet a été largement étudié par les physiciens et les acousticiens. Le LNL permet de reproduire les caractéristiques Force/Vitesse couramment proposées dans la littérature. On distingue sur la courbe (page suivante) la zone d'accrochage (pente positive autour de v=0) et la zone de glissement (hyperbole)[3].

- Excitations diverses : le LNL autorise d'autres interactions d'excitation, tels le pincement ou le raclement.

- Pesanteur : GENESIS est un monde sans pesanteur. Il est possible de simuler une pesanteur en connectant un module <MAT> à un SOL à l'aide d'un LNL. Il suffit de prendre une caractéristique $f_r(DeltaX)$ horizontale – en prenant garde, toutefois, que la force constante soit proportionnelle à l'inertie du MAS si on veut modéliser une gravité uniforme.

[1] Rappelons que la vitesse doit cependant être exprimée en (distance/échantillon) sur l'axe des abscisses de la caractéristique de viscosité. Si les vitesses sont mesurées en (distance/seconde), la pente de la droite doit être z=Z/Fe.

[2] Notons que lorsqu'on substitue un REF par un LNL dans GENESIS, ce dernier est avant d'être modifié par l'utilisateur fonctionnellement identique au REF d'origine. Notons également qu'il n'est pas possible de représenter un BUT avec un LNL : dans la BUT, la force de viscosité est conditionnée à la positions des modules <MAT> connectés, alors que dans le LNL elle est conditionnée à leur vitesse relative. Le BUT et le LNL sont donc bien deux modules complémentaires.

[3] Des interactions de type frottement sont donc possibles dans GENESIS. Il faut cependant noter qu'il est difficile d'obtenir un frottement musicalement satisfaisant. Il semblerait que *seule la finesse de l'interaction gestuelle avec retour de force* peut permettre l'obtention à moindre frais de sons frottés porteurs d'une réelle expressivité (en temps différé, une telle qualité nécessite de passer beaucoup de temps . Cette remarque, incidemment, vaut pour la plupart des excitations continues. Elle ne remet en rien en cause, cependant, le module LNL lui-même.

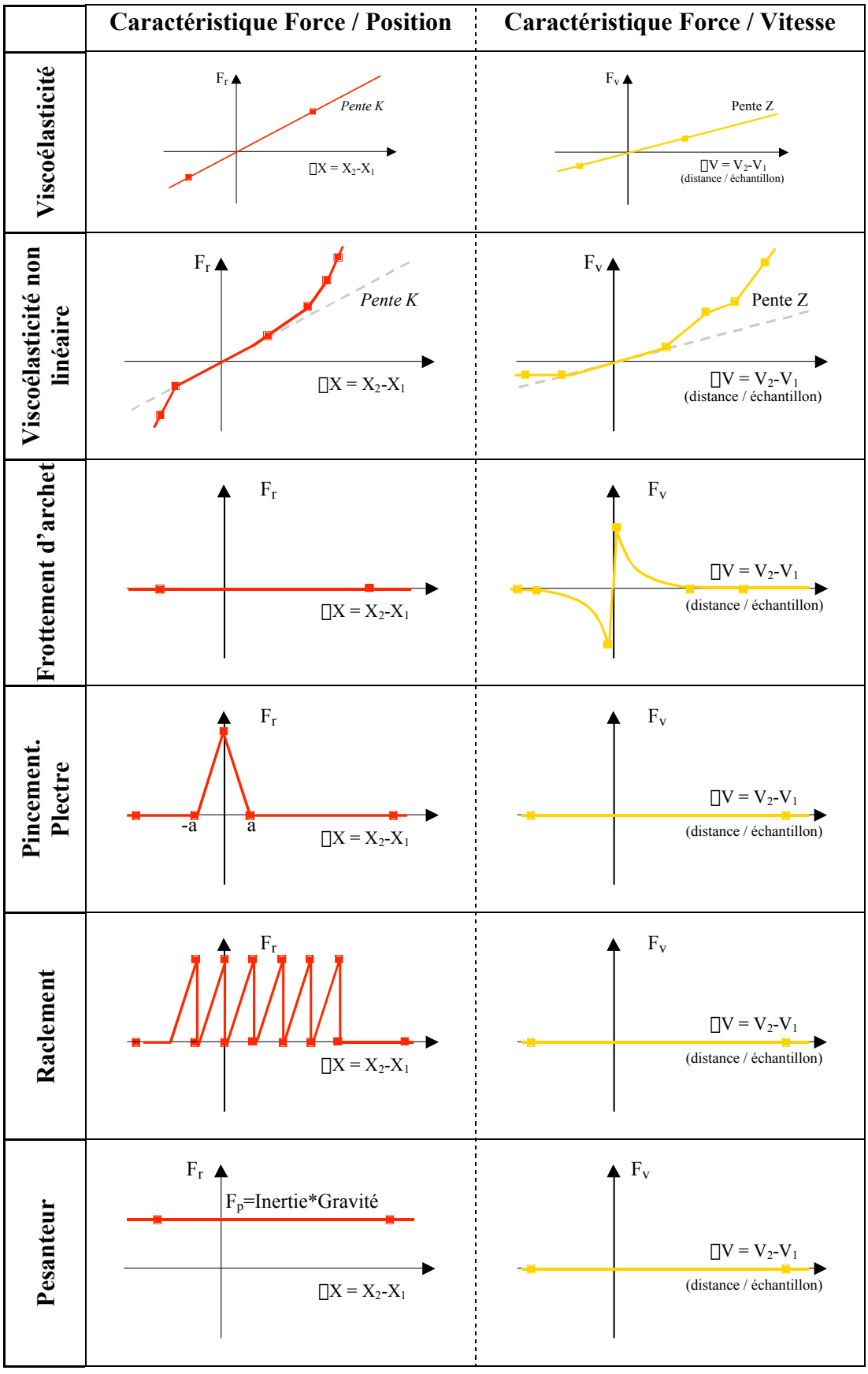


Figure 8 : Quelques exemples d'utilisation du module LNL

Raideur non-linéaire distribuée en X^3 ; module RESNL

L'une des causes essentielles des comportements non-linéaires des objets réels précontraints est l'incidence de l'élongation (longitudinale) sur la *raideur apparente* des modes transverses.

Dans l'annexe C, nous étudions un *filament bidimensionnel* CORDIS-ANIMA précontraint par une tension T et excité transversalement. Nous montrons qu'il est possible de linéariser les forces de raideurs. Au premier ordre, le développement aboutit à un *filament topologique équivalent* linéaire, modèle des ondes transversales qui se propagent dans le filament bidimensionnel. Au second ordre apparaît un terme en X^3 – X étant la position des <MAT> sur l'axe transverse du modèle 2D. Ajoutée aux algorithmes des élasticités du filament topologique équivalent, une telle non linéarité en X^3 induit un effet sonore pertinent pour la perception.

Une série d'expérimentations a montré qu'il est intéressant d'introduire ce terme en X^3 dans les objets topologiques sans qu'il soit davantage nécessaire d'imaginer ces derniers comme une simplification d'un modèle spatial : la sensibilité des timbres au niveau d'excitation qu'il induit est très souvent musicalement pertinente[1].

Nous avons, en conséquence, ajouté un module élastique non linéaire RESNL à la bibliothèque de modules CORDIS-ANIMA. Ce module est défini par trois paramètres K, NL et Z, et par l'algorithme :

$$F_{n,t} = K * D_n \left(1 + NL * D_n^2\right)$$

Dans cette définition, le terme en X^3 apparaît comme une *perturbation* qui serait ajoutée à un module RES. Elle favorise donc la démarche consistant à débuter par un objet linéaire, à substituer dans cet objet les RES par des RESNL de paramètre NL initialement nul, puis à augmenter progressivement, au fur et à mesure des simulations, le terme NL[2].

Incidemment, cette démarche implique que l'introduction du module LNL dans l'interface pose quelques problèmes ergonomiques spécifiques. La résolution de ces problèmes, et plus généralement l'intégration du module RESNL dans GENESIS, ne sont pas apparus comme prioritaire[3]. En conséquence, le module RESNL n'est disponible que dans certaines versions ad hoc du logiciel, et n'apparaît pas dans la version 1.5.

---

[1] En d'autre termes, lorsqu'on veut perturber un modèle initialement linéaire comme nous l'évoquions plus haut, il est intéressant de songer d'abord à un terme en X^3 dans la caractéristique $f_r(DeltaX)$.

[2] Notons que l'influence de la non-linéarité dépend de Dn, c'est à dire du degré d'excitation de l'objet. Plusieurs expériences informelles tendent à montrer que la valeur moyenne de $\left(NL * D_n^2\right)$ au cours de la simulation varie comme l'effet non linéaire perçu. En ce basant sur ce résultat, nous proposons dans l'annexe 1 un procédé permettant d'estimer la valeur de NL garantissant qu'une certaine importance de l'effet non linéaire soit obtenue.

[3] Aujourd'hui, nous considérons que la piste la plus intéressante consisterait à intégrer le REFNL – ou du moins la raideur non-linéaire en X^3 – dans l'ergonomie des modules LNL. Un LNL, alors, pourrait être soit défini point par point, soit par une fonction d'interaction en X^3.

Conclusion sur les non-linéarités

Il est certain que les modules non-linéaires proposés dans GENESIS ne constituent pas une base complète. Par exemple :

- Ils ne permettent pas de mettre en œuvre d'hystérésis ;
- Ils ne permettent pas de conditionner la viscosité d'une liaison à la position relative des deux <MAT> connectés.

Pour autant, les modules BUT et LNL – le cas échéant complétés des modules RESNL – permettent déjà de couvrir une très grande variété de non-linéarités, comme l'ont montré de nombreux travaux d'utilisateurs.

10.5.3 - A propos des modules d'écoute et de mesure

Pour écouter ou *mesurer* un objet CORDIS-ANIMA, il est nécessaire de capter certains des signaux de force ou de position échangés entre modules et de les transformer en signaux sonores.

Deux modules de *mesure* sont utilisés dans GENESIS :

- Le SOX (SOrtie X), un module <LIA> dégénéré (c'est à dire n'ayant qu'un seul point de communication de type L) qui *mesure* la position du <MAT> auquel il est connecté, l'écrit dans un fichier et renvoie une force nulle.
- Le SOF (SOrtie Force), un module <MAT> qui *mesure* la force entrante, c'est à dire la somme des forces exercées par l'ensemble des <LIA> qui lui sont connectés tout en restant immobile sur l'axe de simulation.

Les modules SOX et SOF respectent donc les principes de connexion de CORDIS-ANIMA. Ils disposent de points M ou L et se comportent, lors des simulations, comme n'importe quel module. Quelques remarques s'imposent cependant.

- L'ajout d'un module SOX est indifférent pour la simulation : la mesure qu'il effectue est *non perturbante*. Ce n'est pas le cas du module SOF : immobile, il se comporte pour la simulation comme un module SOL. Seules les forces exercées sur un point fixe sont donc à ce jour mesurables dans GENESIS.
- Les modules de sorties SOX et SOF ne constituent pas une base complète pour la mesure. Ils ne permettent de mesurer que *certains* des signaux échangés entre modules physiques. Avec eux, par exemple, il n'est pas possible de mesurer les signaux de force sortant d'un <LIA> unique.

Compte tenu de ces remarques, nous considérons que les modules SOX et SOF sont appelés à disparaître au profit d'un système de mesure plus complet des objets CORDIS-ANIMA, qui serait en outre intégralement non-perturbant. Une réflexion est en cours au laboratoire.

10.5.4 - A propos des modules d'entrée (et d'interaction)

Les deux modules utilisés sont les symétriques des modules SOX et SOF :

- Le module ENX (Entrée Position) est un <MAT> dont la position est déterminée par l'extérieur (fichier ou TGR©, suivant que la situation est temps réel ou non) et dont la force est, le cas échéant, redirigée vers le TGR©.
- Le module ENF (Entrée Force) est un <LIA> dégénéré qui exerce une force déterminée par l'extérieur (fichier ou TGR©), et qui le cas échéant renvoie vers le TGR© la position du <MAT> connecté.

Deux remarques s'imposent concernant ces modules :

- Dès lors que le recours au moteur temps réel est exclu, il s'agit de modules *d'entrée* et non d'interaction.
- Par ailleurs, nous verrons au chapitre suivant que nous avons exclu de GENESIS toute possibilité de *jeu* autre qu'avec les conditions initiales.

En conséquence, GENESIS 1.5 ne fait pas apparaître ces modules. Ils sont réservés, comme la non linéarité en X3, à quelques versions ad hoc, et particulièrement :

- *G1.40b GesteFourcade - 07-00* : implémentation de deux flux d'entrée dynamique (position et force) pour travail de finalisation de la thèse de Patrick Fourcade.
- *G1.41a TempsRéel - 04-01* : connexion au moteur temps réel pour expérimentations.

10.5.5 - A propos des modules fonctionnels de modification dynamique paramétrique et structurelle

CORDIS-ANIMA définit enfin la notion de module de modification paramétrique (MMP) et de module de modification structurelle (MMS), appelés *modules fonctionnels* dans la mesure où ils opèrent dans le domaine du signal[1].

De façon générale, les modules MMS et MMP visent à permettre la transformation d'un objet en un autre dans le temps de la simulation. Ils semblent à ce titre tout à fait intéressants pour la création. Pour plusieurs raisons, nous avons cependant décidé de ne pas les intégrer à GENESIS.

Tout d'abord, l'implémentation de ces modules au sein des moteurs de simulation n'est pas à ce jour effectuée et de plus il se pourrait que leur principe soit redéfini dans un proche avenir[2].

---

[1] Rappelons :

- Les modules de modification structurelle (MMS) permettent de modifier une structure en cours de simulation (déplacement dynamique d'une connexion, ajout ou suppression de module).
- Les modules de modification paramétriques (MMP) permettent de modifier dynamiquement un paramètre.

[2] Tels qu'ils sont définis, les modules MMS et MMP s'opposent au principe *d'expérimentabilité* qui est à la base de CORDIS-ANIMA (Chapitre 1 paragraphe 1.3.2). Par exemple, lorsqu'un MMP commande la raideur d'un module REF, l'énergie potentielle du REF est modifiée sans que cela ne puisse être perceptible du côté *commande*. Pour satisfaire au principe *d'expérimentabilité*, il serait nécessaire que l'énergie introduite dans le REF (côté effet) soit soustraite du coté commande. Une réflexion formelle a été initiée afin de déterminer si la mise en oeuvre de *quadripôles physiques* en lieu et place des actuels MMS apporterait une réponse adéquate à ce type de problème.

Ensuite, il faut noter qu'il y a un risque à leur mise en œuvre dans la mesure où ils modifient de manière non physique (i.e. : de manière arbitraire) l'état énergétique d'un modèle. Il existe presque toujours des solutions permettant de réaliser la modification souhaitée à l'aide des modules de base de CORDIS-ANIMA.

Ces solutions *physiques* sont en général préférables. D'une part, elles permettent la génération de phénomènes plus satisfaisant (i.e. plus physiques ou *plausibles*). D'autre part, elles sont plus rigoureuses et satisfaisantes du point de vue de la démarche de modélisation[1]. Nous considérons, de fait, que l'intégration dans GENESIS des modules MMP et MMS nuirait au développement d'une *pensée physique* que nous voulons prééminente en introduisant dans le logiciel des notions directement issues du traitement du signal.

Ainsi, l'absence à ce jour des modules fonctionnels dans GENESIS est tout à fait conforme d'une part à nos objectifs quant à l'approche de la création et d'autre part à la démarche incrémentale adoptée pour nos travaux.

10.5.6 - Panorama résumé des modules CORDIS-ANIMA implémentés dans GENESIS

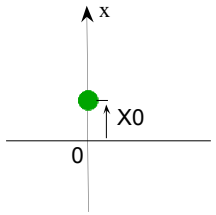
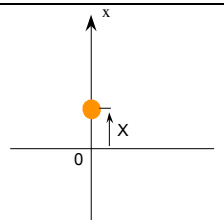
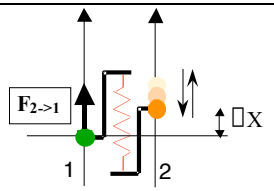
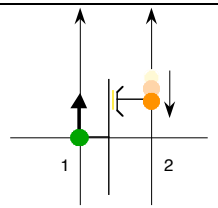
Nous terminerons ce chapitre en proposant un aperçu synthétique de l'ensemble des modules CORDIS-ANIMA disponibles dans GENESIS.

Nous présentons d'abord les 8 modules physiques et les 2 modules de sortie sonore qui figurent dans la version 1.5 du logiciel actuellement diffusée. Les objets CORDIS-ANIMA conçus dans GENESIS 1.5 sont *exclusivement* constitués de ces modules. Figurent ensuite les 2 modules d'entrée ou d'interaction gestuelle et le module non linéaire en X^3 qui ne sont proposés que dans des certaines versions de GENESIS.

[1] Un exemple (voir l'Annexe B pour des compléments) : Comme nous l'avons indiqué plus haut, il n'est pas possible d'appliquer un geste de précontrainte à un modèle *topologique* de corde. Pour obtenir avec un tel modèle un effet évoquant la mise sous tension d'un corde réelle (de violon par exemple), il faut avoir recours à l'artifice de modules MMP et de contrôler dynamiquement les raideurs des <LIA>. Celà fonctionne, mais ce n'est pas la « bonne » solution. Pour rester dans le cadre d'une *pensée physique* naturelle, garantir la cohérence des phénomènes générés et par voie de conséquence les possibilités expressives, il est de loin plus intéressant d'implémenter un modèle *spatial* sur lequel le geste de précontrainte acquière une signification directe.

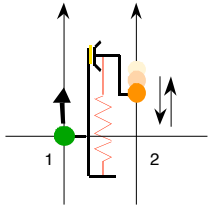
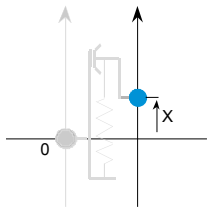
Notons que cela n'est pas possible à ce jour dans GENESIS puisque c'est la version *topologique* de CORDIS-ANIMA qui est utilisée, pour les raisons qu'on sait. Une *perspective* à moyen ou long terme pour le logiciel tient au développement des spatialités plus complexes.

Modules linéaires élémentaires

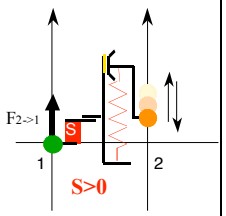
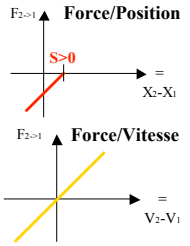
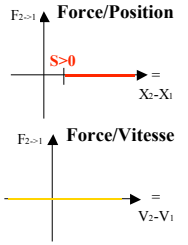
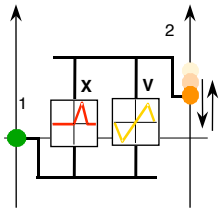
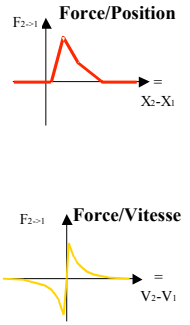
	Représentation fonctionnelle	Comportement physique modélisé	Paramètres	Conditions initiales	Algorithme	Commentaire
SOL		Immobilité. terre, **SOL**.	Aucun	X_0 Pos. initiale	$X_{n+1}=X_n=X_0$	Le module SOL est le *point fixe* des objets CORDIS-ANIMA. On peut aussi l'envisager comme une inertie infinie (par exemple : la terre).
MAS		**MAS**se, Inertie	M : inertie (jamais nulle)	X_0 Pos. Initiale V_0 Vit. initiale	$X_{n+1}=2*X_n-X_{n-1}+\frac{1}{M}F_n$	Le module MAS est le plus courant dans les objets CORDIS-ANIMA.
RES		**RES**sort, Elasticité	K : constante de raideur	Aucune	$F_{2-1}=K\cdot\Delta X=K\cdot(X_2-X_1)$	Les modules RES et FRO ont un intérêt pour la pédagogie. Ils sont rarement mis en oeuvre pour la modélisation, mais restent utiles lorsqu'une liaison uniquement visqueuse ou élastique est nécessaire.
FRO		**FRO**ttement, Viscosité	Z coefficient d'amortissement	Aucune	$F_{2-1}=Z\cdot V=Z\cdot(V_2-V_1)$	

Modules linéaires intégrés

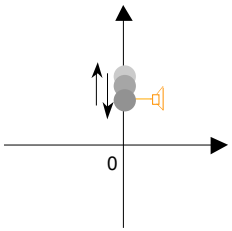
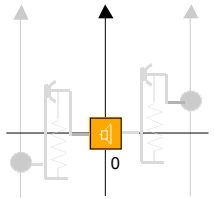
Les modules intégrés résultent de l'assemblage de plusieurs modules élémentaires au sein d'un seul module.

	Représentation fonctionnelle	Comportement physique modélisé	Paramètres	Conditions initiales	Algorithme	Commentaire
REF		**RE**ssort **F**rottement, Visco-élasticité	K : constante de raideur Z : coefficient d'amortissement	Aucune	$F_{2/1} = K/\!/X + Z/\!/V$ $= K/\!/(X_2 - X_1) + Z/\!/(V_2 - V_1)$	Le REF résulte de l'intégration d'un RES et d'un FRO. C'est le module d'interaction le plus courant dans les objets CORDIS-ANIMA.
CEL	La cellule. 	**CEL**lule, Oscillateur mécanique élémentaire. Inertie dans un milieux.	M : inertie K : constante de raideur Z : coefficient d'amortissement	Pour le pt M mobile : X_0 :Pos. Initiale V_0 :Vit. Initiale	$X_{n+1} = 2*X_n - X_{n-1} + \frac{K*X_n + Z*V_n + F_n}{M}$	La cellule CEL est composée d'un MAS, d'un SOL et d'un REF. Seul le MAS mobile est accessible et peut recevoir des connexions. Un CEL est un oscillateur mécanique élémentaire. C'est le plus petit objet oscillant dans l'univers CORDIS-ANIMA. Le CEL sert également à modéliser la viscosité et l'élasticité de milieu.

Modules non-linéaires ; Deux modules non-linéaires sont accessibles dans la version 1.5 de GENESIS.

	Représentation fonctionnelle	Comportement physique modélisé	Paramètres	Algorithme et caractéristique Force / Position et Force / Vitesse		Commentaire
BUT		**BUT**ée viscoélastique. Non linéarité conditionnée à la position. Contact, non-pénétrabilité.	K : constante de raideur Z : coefficient d'amortissement. (K et Z expriment la viscoélasticité pendant le contact) S : seuil (exprime la condition de contact)	**Si X_2-X_1 <= S :** $F_{2\to1} = K\cdot(\Delta X - S) + Z\cdot\Delta V$ $= K\cdot(X_2 - X_1 - S) + Z\cdot(V_2 - V_1)$ (Interaction visco-élastique)		Le BUT est une liaison *orientée*, dissymétrique qui modélise le *contact* entre deux objets. Il permet notamment de *percuter* une structure. Dans la représentation fonctionnelle (à gauche), le module MAS vient ainsi percuter le module SOL. Inséré en grand nombre au sein d'une structure oscillante, il permet d'obtenir des *structures maracassées*..
				Si X_2-X_1 > S : $F_{2\to1} = 0$ (Pas d'interaction)		
LNL		**L**iaison **N**on **L**inéaire. Interaction non linéaire largement paramétrable	Deux courbes caractéristiques $F1_{2\to1} = f1(\Delta X)$ et $F2_{2\to1} = f2(\Delta V)$ définies point par point et par des interpolations : - linéaires, - hyperboliques - ou spline.	A1 : repérer DeltaX dans le premier tableau de point. A2 : Calculer F1, compte tenu de l'interpolation choisie pour ce segment. B1 : repérer DeltaV dans le second tableau de point. B2 : calculer F2 compte tenu de l'interpolation choisie pour ce segment. C : Additionner F1 et F2		Le LNL permet d'exprimer de nombreux types d'interaction : frottement, pincement, pesanteur etc. Inséré au cœur d'une structure, il permet d'exprimer une visco-élasticité non-linéaire.

Modules de sortie sonore (mesure)

	Représentation fonctionnelle	Description générale	Paramètres	Conditions initiales	Algorithme	Commentaire
SOX		**SO**rtie **X** Capture de position	Fichier son Numéro de la voix sonore destinataire	Aucune	**Pour la simulation :** aucun effet (mesure non perturbative) **Pour l'extérieur :** capture de la position du <MAT> auquel le SOX est connecté et enregistrement du signal obtenu dans un fichier.	C'est un <LIA> dégénéré à un seul point M. Lors de l'écoute du signal, le mouvement de la membrane du haut-parleur sera proportionnel à celui du <MAT> connecté. Pour écouter plusieurs structures en même temps, il faut que les amplitudes des différentes structures soient compatibles.
SOF		**SO**rtie **F**orce Capture de force.	Fichier son Numéro de la voix sonore destinataire.	X_0 Pos. Initiale (conservée pendant toute la simulation)	**Pour la simulation :** le SOF se comporte comme un point fixe SOL : $X_{n+1}=X_n=X_0$ **Pour l'extérieur :** capture de la somme des forces exercées par les <LIA> connectés et enregistrement du signal obtenu dans un fichier.	La mesure est perturbante : le SOF est un point fixe pour la simulation. Dans GENESIS, on ne peut pas mesurer de forces hormis celles qui entrent dans un point fixe. Même remarque que pour le SOX quant à la nécessaire compatibilité des échelles d'amplitudes au sein du modèle.

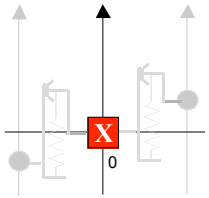
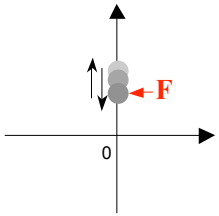
Il convient de noter que les modules SOX et SOF sont appelés à disparaître dans les versions ultérieures de CORDIS-ANIMA et GENESIS au profit de modules de mesure non perturbante. Ces modules de mesure ne répondraient plus aux règles de connexion CORDIS-ANIMA mais permettraient de capter le signal sortant de n'importe quel module.

Modules utilisés dans des versions personnalisées

Nous précisons enfin les 3 modules utilisés dans certaines versions personnalisées de GENESIS.

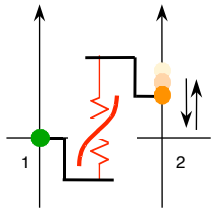
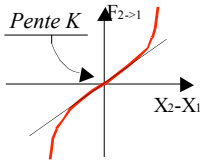
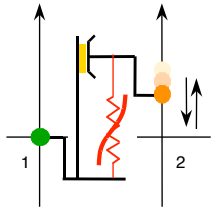
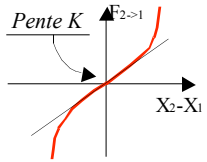
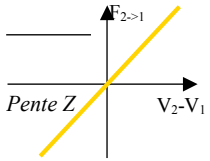
- Modules d'entrée et d'interaction

Les modules d'entrée et d'interaction figurent dans les versions de GENESIS orientées vers l'expérimentation du jeu à temps différé *et* dans les versions expérimentales temps réel avec interaction mutlisensorielle via le TGR©.

	Représentation fonctionnelle	Description générale	Paramètres	Conditions initiales	Algorithme	Commentaire
ENX		**EN**trée **X** Entrée de position et retour de force dans le cas de l'interaction temps-réel	Fichier geste ou touche du transducteur	X0 position du module en l'absence de signaux ou fichier d'entrée.	**Pour la simulation :** la position du module est lue dans le fichier (ou captée sur le transducteur) **Pour l'extérieur (situation temps réel interactive) :** envoi vers le TGR© de la somme des forces exercées par les <LIA> connectés au ENX.	Le module ENX est l'image, dans le modèle, d'une touche du Transducteur Gestuel Rétroactif ou d'un fichier (suivant que la simulation est temps différé ou temps réel.
ENF		**EN**trée **F**orce Entrée de force et retour de position dans le cas de l'interaction temps-réel	Fichier geste ou touche du transducteur	Aucune	**Pour la simulation :** La force est lue dans le fichier (ou captée depuis le transducteur). Nulle si le fichier est vide ou inexistant. **Pour l'extérieur (situation temps réel interactive) :** envoi vers le TGR© de la position du <MAT> auquel le ENF est connecté.	

- Non linéarité distribuée ; module RESNL et REFNL

Les modules RESNL et REFNL ne sont définis que dans certaines versions du logiciel. A l'avenir, ces modules sont appelés à être intégrés, dans l'interface, aux côtés du LNL. Il n'y sera plus fait référence dans la suite du document, à l'exclusion de l'annexe B.

	Représentation fonctionnelle	Comportement physique modélisé	Paramètres	Algorithme et caractéristique Force / Position	Commentaire
RESNL		Raideur non linéaire en X^3 Non-linéarité distribuée	K constante de raideur NL terme non linéaire	$F_{2\to 1} = K \cdot X \cdot (1 + NL * X^2)$ $= K \cdot (X_2 - X_1) * \left[1 + NL * (X_2 - X_1)^2\right]$ 	Le RESNL est *raideur non linéaire* qui fait apparaître un terme en X^3 (cube de la position relative des deux <MAT> connectés). Le RESNL permet de modéliser dans l'univers topologique les comportements non-linéaire apparaissant dans les structures précontraintes du fait de leur élongation. Lorsqu'il est utilisé pour remplacer *l'ensemble* des RES d'un modèle initialement linéaire (non linéarité distribuée), il induit des effets souvent pertinents.
REFNL		Raideur non linéaire en X^3 + viscosité	K : raideur NL : terme non linéaire Z : viscosité	  $F_{2\to 1} = K \cdot X \cdot (1 + NL * X^2)$ $+ Z * V$	A l'image du REF, le REFNL résulte de l'intégration au sein d'un même module d'un RESNL (raideur non linéaire) et d'un FRO (viscosité simple).

Chapitre 11

ORGANISATION GENERALE DE GENESIS

Ce second chapitre introduit l'organisation générale que nous avons mise en place pour GENESIS. Il aborde successivement :

- Les différents modes que nous avons définis et la raison pour laquelle nous nous sommes particulièrement attachés à la conception du mode lutherie.
- Les différents types de données cohabitant dans l'environnement GENESIS, ainsi que leurs liens de dépendance et leur organisation physique (arborescence des comptes utilisateurs).
- Les propriétés fondamentales de l'interface GENESIS et particulièrement de l'interface du mode lutherie : style d'interaction, organisation des fenêtres à l'écran, menus, etc.

11.1 - Les différents modes et les fonctionnalités de simulation

11.1.1 - lutherie, jeu, composition, gestion : les modes d'une version idéale de GENESIS

Le chapitre II de la partie V a montré comment le processus de création s'organise en deux phases : la phase de lutherie et la phase de jeu. Cette dualité marquée tend à ce que plusieurs *modes* apparaissent dans GENESIS. C'est l'option que nous avons retenue.

A chaque *mode* de GENESIS correspond :

- Une phase du processus de création ;
- Un type de données manipulables ;
- Un ensemble d'actions possibles sur ces données ;
- Une *manière de penser* et un savoir-faire spécifique pour l'utilisateur – on pourrait aller jusqu'à dire un *métier* ;
- Des conditions à respecter pour pouvoir changer de mode.

Il est certain que le choix de plusieurs modes tend à segmenter les tâches de l'utilisateur, à imposer une organisation à son travail. Il convient donc de ne pas multiplier les modes et de les choisir de façon judicieuse. Par exemple, il serait malvenu que deux modes différents soient définis pour l'édition de la structure du réseau <MAT>/<LIA> et l'édition des paramètres. Nous considérons qu'une organisation optimale de GENESIS ferait apparaître quatre modes : les modes *lutherie* et *jeu* tout d'abord, dont les rôles respectifs se déduisent de l'analyse du processus de création menée dans la partie V, les modes *gestion* et *composition* ensuite.

Le mode *lutherie* se caractérise ainsi :

- Il est associé à la phase de lutherie du processus de création et son utilisation aboutit à la conception d'un objet CORDIS-ANIMA.
- Il permet la manipulation du langage CORDIS-ANIMA.
- Il regroupe l'ensemble des fonctionnalités d'édition de ces objets.
- Il repose sur la maîtrise de CORDIS-ANIMA et la pensée physique.

Les caractéristiques du mode jeu seraient les suivantes :

- Il est associé à la phase de jeu du processus de création et son utilisation aboutit à la spécification d'une séquence gestuelle à appliquer à l'objet qui, le cas échéant, comprend certaines parties à exécuter en temps réel.
- Les données qui y sont manipulées sont essentiellement des signaux gestuels – c'est à dire les signaux appliqués aux modules d'entrée et d'interaction ou des actions gestuelles élémentaires appliquées aux modules physiques de l'objet.
- Il regroupe l'ensemble des fonctionnalités d'édition de ces signaux.
- Il repose sur une pensée temporelle ou évènementielle – en tout cas une pensée dans laquelle le temps est explicite.

Le mode composition regrouperait alors les tâches de haut niveau qui n'ont été qu'évoquées à la fin de la partie V. De façon très générale, on pourrait le définir en disant qu'il correspond à une phase d'intégration et de finalisation des travaux menés en mode lutherie et en mode jeu. Avec le mode composition, l'utilisateur ne modifierait plus les objets ou les signaux gestuels mais s'intéresserait à une « composition » de l'ensemble de ces données à un plus haut niveau d'abstraction et le cas échéant manipulerait d'autres types de données tels des partitions pour un instrumentiste. Le mode composition reposerait donc sur une pensée globale, dans laquelle interviennent à la fois des considérations physiques (par la composition de structures à un haut niveau d'abstraction) et des considération temporelles (les gestes pouvant se succéder dans le temps, de même que les objets). Ses principes et ses fonctionnalités sont encore mal définis aujourd'hui et une réflexion approfondie les concernant doit être entreprise.

Enfin, nous considérons qu'un quatrième mode devrait figurer dans GENESIS pour permettre l'organisation physique de l'ensemble des données de l'utilisateur. Les principes du mode Gestion pourraient s'inspirer des systèmes de fichiers. Dans ce mode, l'utilisateur pourrait naviguer dans les données, les déplacer ou supprimer, accéder à des catalogues d'objets (le cas échéant par le réseau Internet), procéder à des simulations en tâche de fond, etc.

11.1.2 - La simulation, un paradigme central

On pourrait songer à créer un cinquième mode dont le rôle serait de regrouper les fonctionnalités nécessaires à la simulation. Réaliser une simulation, en effet, c'est passer dans une phase particulière du processus de création, qui correspond à une autre *manière de penser* : il ne s'agit plus d'éditer un quelconque type de données mais d'observer les phénomènes générés.

Il apparaît cependant que le paradigme de la simulation est transversal à l'ensemble du processus de création : quel que soit le mode utilisé, la simulation est nécessaire pour valider les actions réalisées. En mode lutherie, par exemple, la simulation permet de valider – ou d'invalider – l'objet conçu. En mode jeu, il s'agit d'estimer l'intérêt des gestes édités.

En conséquence, nous avons décidé de ne pas créer un mode spécifique pour la simulation, mais *d'autoriser les simulations dans chaque mode* avec, le cas échéant, des spécificités quant à la façon dont les données recueillies sont portées à la connaissance de l'utilisateur. Dans les versions futures de GENESIS, qui plus est, les deux systèmes de simulation temps réel et temps différé devraient être accessibles dans chaque mode.

<u>En résumé: l'organisation idéale des modes</u>

L'organisation générale de l'environnement que nous venons de décrire est résumée sur la figure 9.

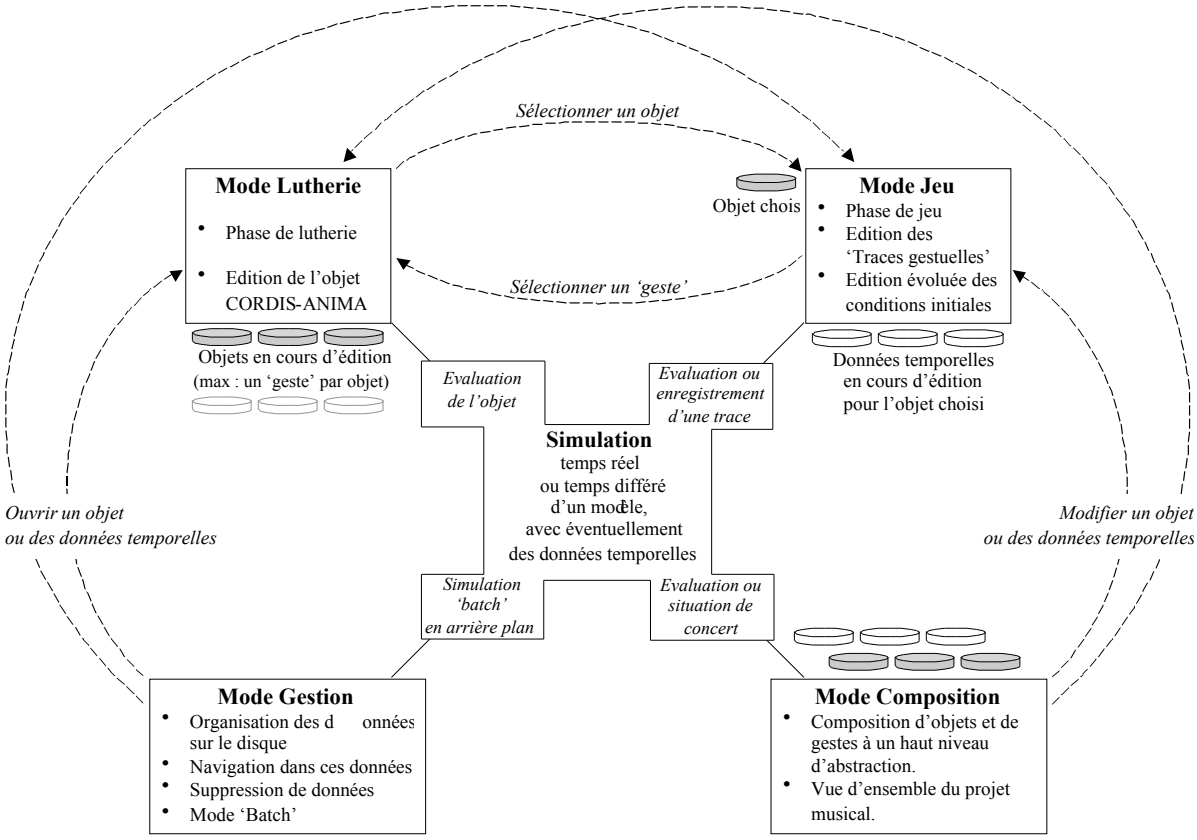


Figure 9 : modes de GENESIS et conditions de passage entre modes
(les conditions de passage sont les arcs en pointillé).

11.1.3 - L'implémentation réalisée : restriction au mode lutherie

La coexistence des modes Lutherie, Jeu, Composition et Gestion constituerait une situation optimale, vers laquelle les développements futurs de GENESIS devraient tendre. La situation effective de GENESIS 1.5 en diffère cependant encore largement.

Le mode jeu, tout d'abord, n'existe que dans certaines versions spécifiques ou personalisées de GENESIS et qui plus est à l'état embryonnaire[1]. Dans la version 1.5, les fonctionnalités de jeu sont limitées à la spécification des conditions initiales de l'objet[2] et sont intégrées au mode Lutherie. De fait, la conception d'un mode Jeu séparé suppose une réflexion approfondie sur plusieurs sujets tels : les modules d'entrée et d'interaction, la notion de signaux évènementiels et la manière dont ces signaux pourraient être traduits dans l'environnement physique des objets CORDIS-ANIMA, la notion d'Action Gestuelle Elémentaire, les problèmes inhérents à l'édition de la trace, etc. Cette réflexion est aujourd'hui en cours au laboratoire ; elle réunit les trois projets principaux que sont GENESIS, MIMESIS et TELLURIS.

Le mode *composition* est à ce jour le moins avancé. L'analyse des tâches que nous avons conduite à la partie V ne fait qu'évoquer la possibilité de tâches plus abstraites et complexes. Si on comprend sa nécessité, la réflexion concernant les fonctionnalités et objectifs du mode composition sont encore à l'état embryonnaire.

A ce jour, des fonctionnalités élémentaires de gestion des données et du compte utilisateur sont intégrées au mode lutherie. Les bases conceptuelles du mode Gestion spécifique sont cependant d'ores et déjà conçues. Elles seront implémentées dans leur ensemble lorsque la complexité et surtout la diversité des données à manipuler sera plus importante et/où lorsque le réseau des utilisateurs de GENESIS sera suffisamment développé pour qu'il soit intéressant d'échanger des données par le réseau Internet et d'accéder à des catalogues distants.

Ainsi, *la version 1.5 de GENESIS ne dispose que du seul mode lutherie*. C'est sur ce mode qu'ont porté l'essentiel de nos efforts. Sa conception, en effet, nous est apparue comme prioritaire. C'est que la phase de lutherie est préalable à l'ensemble des autres phases du processus de création. Elle est de plus *suffisante* pour qu'un processus de création complet soit mis en œuvre – notamment par le principe de la composition de structures. Enfin elle *concentre* une grande partie des innovations qu'introduit le paradigme de la simulation interactive et multisensorielle d'objets physiques dans les processus de la création musicale.

---

[1] Notons que la version 1.2 de GENESIS disponible au début de nos travaux faisait apparaître un mode Jeu. Ses fonctionnalités, cependant, étaient à la fois réduites, mal définies et mal organisées : elles ne satisfont ni le critère *d'utilisabilité* ni le critère *d'utilité* – et encore moins la nécessité d'apprentissage. Elles illustrent combien il est préférable de concevoir un logiciel réduit mais bien pensé plutôt qu'un ensemble étendu de fonctionnalités difficilement utilisables. Face à cette situation, l'une de nos premières décisions fut de supprimer de la version standard de GENESIS le mode Jeu, en attendant qu'il soit entièrement redéfini. Un mode Jeu embryonnaire reste cependant accessible aux utilisateurs avertis et désireux de l'expérimenter. Ce fut le cas, notamment, de quelques stagiaires et doctorants.

[2] Nous verrons que GENESIS propose un soutien élémentaire pour le contrôle des déclencheurs et des dates des évènements qu'ils génèrent.

11.2 - Fichiers et comptes utilisateurs

Nous précisons dans ce paragraphe :

- Les différents types de données et les formats de fichiers qui cohabitent dans GENESIS ;
- Les liens de dépendances que nous avons considérés entre ces données
- L'organisation physique des fichiers sur disque, c'est à dire l'arborescence des comptes utilisateurs.

11.2.1 - Les différents types de données ; les différents formats de fichiers

Au sein de l'environnement GENESIS cohabitent les données relatives aux objets physiques, les données relatives aux gestes, les données consécutives à des analyses, les données résultant des simulations et enfin des images.

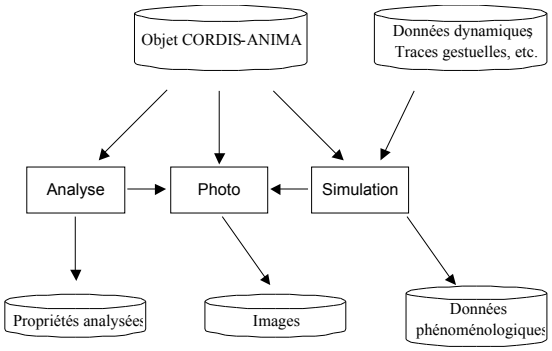


Figure 10 :
Différents types de données dans GENESIS.

Nous en proposons un rapide aperçu dans ce qui suit[1].

<u>Première approche des fichiers objet .top ; fichier .lst</u>

Les premiers types de données que nous envisageons son ceux qui décrivent les *objets* CORDIS-ANIMA.

Paramètre. Un *objet*, dans le monde réel, est appréhendé à la fois comme une *structure* ou une géométrie et comme une matière. Pour soutenir la métaphore dans GENESIS, nous avons préféré inclure dans les fichiers définissant les objets l'ensemble des paramètres des modules[2].

Conditions initiales. Les conditions initiales constituent à la fois une propriété de l'objet (elles définissent son état standard) et une donnée dynamique élémentaire (elles permettent de l'exciter). Dans le cas des modèles topologiques, cette seconde propriété est encore accentuée. Pour renforcer, à nouveau, la métaphore de l'objet, nous avons préféré mettre l'accent sur la première propriété. Les conditions initiales figurent donc, avec la structure et les paramètres, dans le fichier de définition de l'objet. Notons cependant que ce choix pourra être contourné au sein du futur mode jeu. Il sera alors possible de définir des gestes élémentaires qui, appliqués aux modules avant même que la simulation n'ait débuté, prendront le pas sur les conditions initiales définies en mode lutherie. Cela permettra que

[1] Les fichiers temporaires ou d'initialisation ne sont pas évoqués dans ce paragraphe, qui est consacré uniquement aux données essentielles pour l'utilisateur.

[2] Pour tester une nouvelle configuration de paramètres, il faut donc créer un nouvel objet, de sorte que l'exploration paramétrique, dont nous avons souligné l'importance pour GENESIS (partie V) est en apparence rendue difficile. Il eut été possible de sauvegarder structures et paramètres dans des fichiers séparés, mais alors un objet, pour être défini, aurait nécessité que deux fichiers soient associés et la métaphore que nous évoquions était moins directe.

soient définies, dans des fichiers geste, plusieurs configurations de conditions initiales pour un même objet.

En résumé, l'ensemble des données définissant un objet CORDIS-ANIMA sont, dans GENESIS, enregistrées dans un fichier unique, qui d'ailleurs est prêt à être confié au simulateur.

Deux formats de fichiers coexistent cependant :

- Le format .top est un format binaire propriétaire[1]. Les fichiers .top sont les fichiers centraux du mode lutherie, autour desquels l'ensemble des données s'organisent. Ils codent, en plus de la structure, des paramètres et des conditions initiales de l'objet, de nombreuses autres informations que nous présenterons plus loin (commentaires, Ensembles, Capsules, noms des fichiers sons et des analyses auxquels l'objet a donné naissance, etc.).
- Les fichiers .lst sont des fichiers textuels d'import/export. Leur format a été conçu d'une part pour que la lecture directe et le cas échéant la modification des données soit aisée, d'autre part pour que les fichiers puissent être importés dans un tableur. Dans ces fichiers, les informations concernant chaque module sont séparées par une tabulation, et chaque module est séparé des autres par un retour chariot.

Fichiers gestes

De par leur importance dans le processus de création complet et quand bien même la version 1.5 de GENESIS soit dans son utilisation habituelle limitée au mode lutherie, il nous semble important d'évoquer ici les *fichiers gestes*.

Dans la version de GENESIS qui implémente un mode jeu rudimentaire, les fichiers gestes restent extrêmement simples : ils sont constitués d'une suite de flottants codés sur 32 bits, et ne comprennent qu'une seule et unique voie.

Un tel format est, bien sûr, fort peu satisfaisant. Le format des fichiers gestes et plus profondément le paradigme du geste est appelé à être redéfini, notamment pour que des données évènementielles puissent être spécifiées. Plusieurs travaux sont actuellement en cours en ce sens.

Fichiers sons .echf et .aiff

L'utilisation d'un format standard pour les fichiers sons, donc d'un codage *entier* des échantillons, aurait nécessité que l'amplitude maximale des signaux générés puisse être estimée avant les simulations et que les échantillons captés par les modules SOX ou SOF soient normés au fur et à mesure de leur génération. Un choix malvenu pour le coefficient de normation aurait alors pu rendre caduque une simulation – parfois très longue – en induisant soit une saturation, soit au contraire des signaux d'amplitude trop faible. Nous avons donc préféré définir un format propriétaire : le format .echf[2].

[1] Notons que plusieurs formats .top se sont succédés au fur et à mesure des évolutions du logiciel. Une compatibilité ascendante est bien sûr conservée : les fichiers au format 1.2 restent lisibles dans les versions ultérieures du logiciel. Les formats ultérieurs (1.4 et 1.5) partagent la même structure d'en tête et la même architecture. Une compatibilité descendante est ainsi assurée : un fichier au format 1.5 reste lisible sur la version 1.4 du logiciel – les informations nouvelles présentes dans le fichier (groupes, capsules, référence vers les sons….) n'étant pas prises en compte lors de la lecture.

[2] Un ajustement automatique de la sensibilité était envisageable. Nous l'avons rejeté, considérant qu'il était préférable que les données résultant des simulations restent brutes.

Dans les fichiers .echf, les échantillons sont codés sur 32 bits en virgule flottante et les différentes voies étant entrelacées. Les fichiers .echf n'ont pas d'en-tête : c'est l'objet qui leur a donné naissance qui connaît leur structure. Ils ne sont pas accessibles à l'utilisateur en dehors de GENESIS ; lorsqu'il veut disposer d'un fichier son, l'utilisateur est appelé à *exporter* le résultat des simulations.

Le format utilisé pour les sons exportés est le format standards AIFF. Dans les fichiers .aiff générés par GENESIS, la fréquence d'échantillonnage est fixée à 44100 Hz. L'utilisateur peut, par contre, suivant son habitude, choisir un codage entier sur 16 ou 24 bits.

Fichier d'état .stt

Il est intéressant de pouvoir redémarrer une simulation à un instant donné de l'histoire de l'objet. Cela permet, par exemple, de poursuivre l'enregistrement d'un son à partir de l'endroit où la simulation précédente s'était arrêtée. Pour qu'une telle action soit possible, il faut que l'état de l'objet à l'instant considéré soit sauvegardé. C'est la fonction des fichiers d'état .stt.

L'état d'un objet est défini par les positions instantanées et retardées de chacun de ses points M sur l'axe de simulation. La structure des fichiers .stt s'en déduit : il s'agit d'un ensemble de doubles (réels virgule flottante 64 bits), dans lequel sont listées les positions et positions retardées de chacun des points M de l'objet. Les modules y sont ordonnés par type (SOL, MAS, CEL, SOF, ENX). Comme les fichiers .echf, les fichiers .stt n'ont pas d'en-tête : leur structure est déterminée et connue par l'objet qui a été simulé.

Fichiers d'analyse

Dans le chapitre 17 de la partie VI, nous présenterons la fonctionnalité d'analyse modale de GENESIS. Une telle analyse, appliquée à un ensemble de modules d'un objet, génère une liste de modes. Chaque mode est défini d'une part par sa *raideur* et sa *viscosité* caractéristiques, dont on peut déduire la fréquence et le temps d'amortissement, d'autre part par sa *déformée*. La structure des fichiers .ana qui enregistrent ces données est à l'image de ces différentes données. Notons que comme précédemment les fichiers .ana n'ont pas d'en-tête ; c'est l'objet qui a été analysé (ou plutôt la partie de l'objet qui a été analysée) qui permet de les décoder.

Fichier image .rgb

Enfin, parmi les différents types de fichiers présents dans l'environnement GENESIS figurent des fichiers graphiques au format standard .rgb. Ils sont générés dès lors que l'utilisateur veut créer une image à partir de l'un de ses objets.

Le tableau proposé page suivante suivant résume tous ces types de fichier.

Causes	Fichier objet GENESIS	.top		Contient l'ensemble des données définissant un objet : structure, paramètres, conditions initiales, groupes, commentaires, ainsi que les références des fichiers dépendant de l'objet : son, analyse, image… Format binaire propriétaire.
	Fichier objet import/export	.lst	Import Export	Format pour l'import/export d'objets CORDIS-ANIMA. Format texte, séparateur tabulation. Contient uniquement la structure, les paramètres et conditions initiales de l'objet.
	Fichier geste	.gst		Fichier gestes, qui *seront* générés dans le futur mode jeu. Contenu et format en cours de définition.
Phénomènes	Fichier état	.stt		Conserve l'état de l'objet à un instant donné. Permet de reprendre une simulation à cet instant.
	Fichier son	.echf		Fichier son format flottant entrelacé (32 bit / ech.) généré lors des simulations. Conservé, il permet de poursuivre une simulation antérieure.
	Fichier son	.aiff	Export	Fichier son format standard AIFF (16 bits / ech). Permet dans GENESIS d'économiser l'espace disque, en perdant toutefois la possibilité de poursuivre une simulation. Format d'exportation des sons
Analyse	Fichier d'analyse modale	.ana		Fichier contenant les résultats de l'analyse modale d'une partie d'un objet.
Divers	Fichier image	.rgb	Export	Exportation de « photos » d'un objet en cours de conception ou de simulation.

Figure 11 : les différents formats de fichiers dans GENESIS.

11.2.2 - Objet, analyses et phénomènes : causalité et versionnage

Conservation du lien entre cause et effet

Dans un modeleur-simulateur tel que GENESIS, il existe par nature un lien entre les différentes données. Les phénomènes, par exemple, résultent de la simulation d'un objet CORDIS-ANIMA, sur lequel un geste a le cas échéant été effectué.

Nous avons décidé de conserver et d'expliciter les liens de causalité entre les différents types de données au sein de GENESIS (figure 12). Les résultats des simulations –notamment les sons – et des analyses ne sont pas accessibles directement par l'utilisateur. Ils n'ont d'existence qu'à travers l'objet qui leur a donné naissance et dont ils sont, en quelque sorte, des propriétés. Ainsi, on n'écoute pas un son dans GENESIS ; *on écoute un objet* ou, à minima, l'un des sons qu'il peut produire.

GENESIS est donc organisé de telle sorte qu'à tout instant l'objet (donc le fichier .top) connaisse les phénomènes et les analyses qui lui sont relatifs. Cette correspondance objet / phénomène est, par ailleurs, archivée dans les fichiers .top.

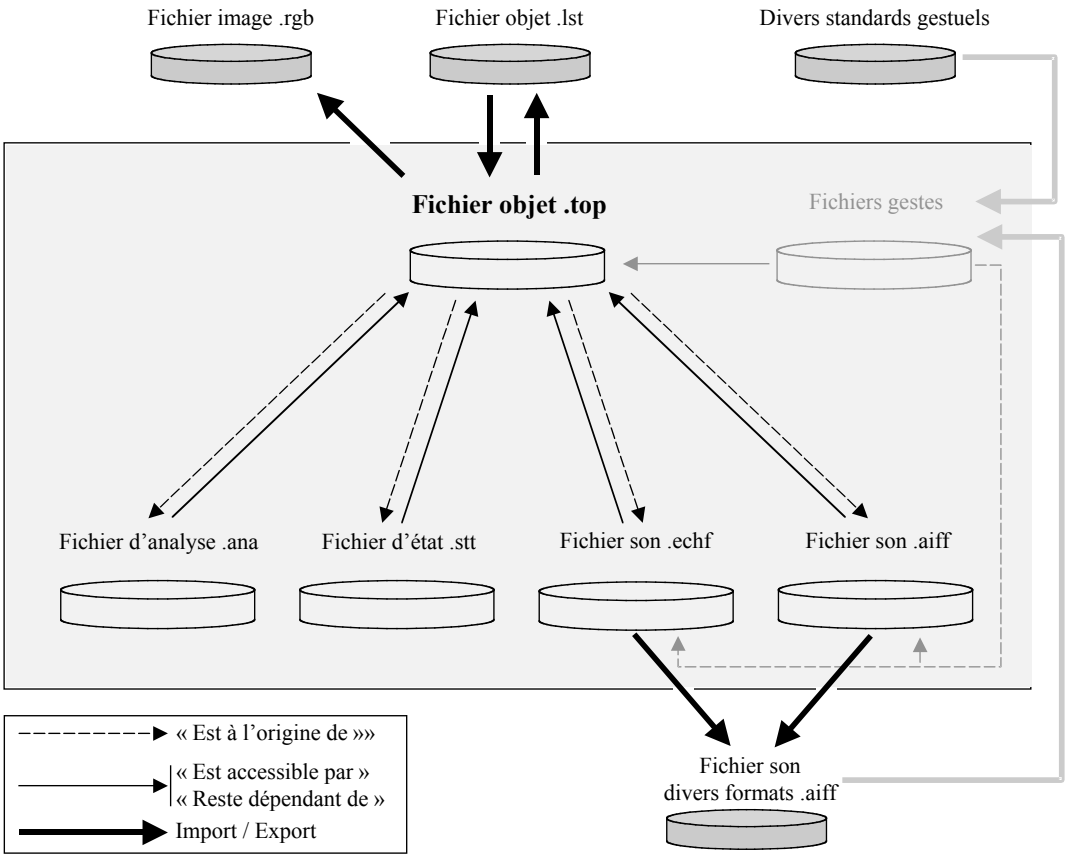


Figure 12 : lien causal entre les différents fichiers ; exportation.

Exportation

Pour autant, comme nous en avons discuté à la partie V, il est essentiel que GENESIS puisse échanger des données avec son environnement. Les fonctionnalités d'exportation y pourvoient. Exporter des phénomènes ou des analyses c'est dans GENESIS précisément :

- Rendre ces données accessibles pour d'autres logiciels – notamment en choisissant un format d'exportation, un nom, un répertoire de destination, etc.
- Et surtout rompre le lien causal qui existait entre ces données et l'objet qui en était l'origine.

Ainsi, exporter un son c'est transformer le son *d'un objet* (contrôlé, le cas échéant, par un geste) en *objet sonore* et lui conférer une *vie autonome*.

Série de structures et démarche incrémentale

Compte tenu des choix retenus quant à la conservation du lien causal, toute modification d'un objet peut entraîner la péremption des phénomènes et analyses qui lui sont relatifs[1].

Pour les analyses, qui en général nécessitent un temps de calcul raisonnable, cela n'est guère problématique. Dans GENESIS, trois cas sont possibles :

- L'utilisateur peut *protéger* les parties analysées de ses objets contre toute modification ;
- Il peut spécifier qu'une analyse doit être recommencée – en tâche de fond – à chaque fois qu'une modification est susceptible de périmer les données ;
- Enfin, dans tous les autres cas, les analyses sont supprimées dès lors que l'objet (ou la partie analysée) est modifié.

La *démarche itérative* qui conduit l'utilisateur à simuler puis modifier l'objet de façon récurrente implique, par contre, qu'un processus différent soit mis en place pour les données de nature phénoménologique. Le système de versionnage et la notion de série d'objets y pourvoient.

Une version est repérée, dans le nom de l'objet, par un numéro précédé du double point « : ». Les noms des objets – et des fichiers – sont donc de la forme :

Designation:XXX:Commentaire

où XXX est le numéro de version de l'objet[2]

La notion de série d'objets se déduit des choix syntaxiques retenus pour les noms des fichiers. Une série, c'est précisément un ensemble d'objets dont les noms partagent la *désignation* et diffèrent par le *numéro de version* et dont les fichiers sont enregistrés dans le même répertoire. GENESIS propose quelques fonctionnalités pour organiser les séries, accessibles pour l'heure en mode lutherie. Elles permettent notamment :

- D'accéder rapidement à une version quelconque d'un objet déjà ouvert ;
- De renommer l'ensemble des objets d'une série (changer la *désignation*) ;
- De renuméroter l'ensemble des versions des objets d'une série par ordre chronologique de création ;
- De créer une nouvelle série à partir de plusieurs objets d'une ancienne ;
- De créer une série à partir de plusieurs objets quelconques – cela permettant de construire des parcours pédagogiques entre différents objets.

A la notion de série est associé un processus de versionnage automatisé. A tout instant, l'utilisateur peut décider de dupliquer un objet dans un autre, auquel un nouveau numéro de version est attribué. Lorsque l'utilisateur s'apprête à modifier un objet auquel un son est attaché, GENESIS propose soit de supprimer le son, soit de créer une nouvelle *version* de l'objet. Ainsi, le lien causal objet/son est conservé tout au long de l'histoire de l'objet (figure 13).

[1] Il convient de nuancer ce propos : les analyses ne sont relatives qu'à certaines parties des objets. De même, certains sons peuvent ne concerner que des sous objets (résultats de simulations partielles).

[2] En conséquence, il n'est pas possible dans GENESIS d'utiliser le caractère « : » dans les noms de fichiers.

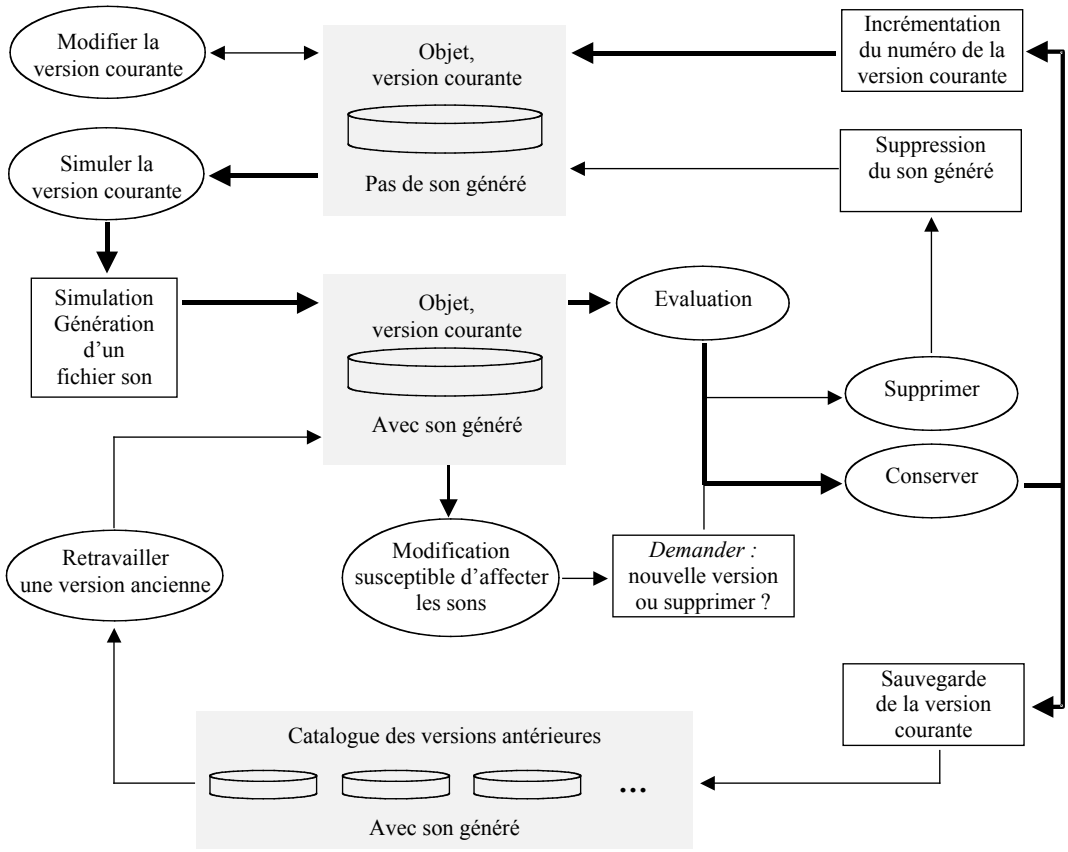


Figure 13 : le processus de versionnage dans GENESIS
La boucle des flèches plus foncées est la boucle typique modification / simulation / évaluation supportant la démarche incrémentale présentée dans le chapitre 8.

11.2.3 - Arborescence des comptes utilisateurs

Deux types d'arborescences étaient envisageables pour organiser les données sur disque :

- L'arborescence par *projet* consiste à ce que *tous* les fichiers relatifs à un ensemble d'objets (typiquement à une série) soient groupés dans un même répertoire. Le nom et l'emplacement de ce répertoire doivent alors être choisis au début du projet.
- L'arborescence simple tend à laisser l'utilisateur organiser l'arborescence de ses fichiers.

L'arborescence projet est assez courante dans les logiciels tels que GENESIS. Nous ne l'avons cependant pas choisie, considérant :

- Qu'elle tend à interdire l'utilisation des fonctionnalités de gestion de fichier du système d'exploitation. En effet, dans une telle arborescence le déplacement d'un fichier risque de nuire à la cohérence des données.
- Qu'elle tend à laisser accessibles les phénomènes et analyses que pourtant nous voulons cacher, puisqu'ils sont stockés à proximités des fichiers .top enregistrant les objets.
- Qu'elle limite les facilités d'archivage et d'envoi de données. Pour archiver un compte utilisateur GENESIS, ou encore lorsqu'il s'agit d'envoyer (par le réseau Internet par

exemple) certaines données à un utilisateur distant, il n'est à priori pas nécessaire d'envoyer les phénomènes et analyses : ceux-ci peuvent être reconstitués plus tard. Il suffit, en fait, d'envoyer ou d'archiver le fichier .top correspondant.

Nous avons donc préféré archiver les fichiers .top et les phénomènes et analyses séparément. La figure 14 présente l'arborescence retenue pour les comptes utilisateurs.

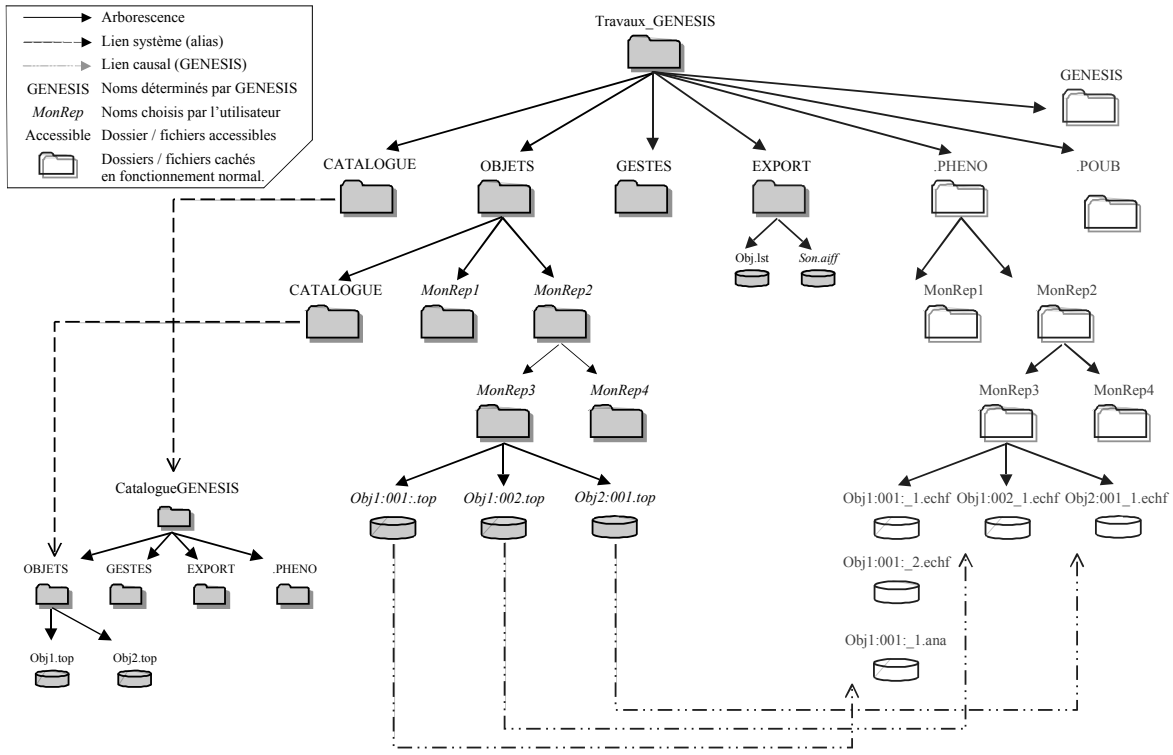


Figure 14 : arborescence des comptes utilisateurs

Dans le répertoire GENESIS figure l'ensemble des fichiers d'initialisation et de préférences.

Dans le répertoire STRUCTURES, l'utilisateur est appelé à enregistrer les objets .top avec l'arborescence qui lui convient. Rappelons que les objets d'une même série doivent être placés dans le même répertoire.

Dans le répertoire GESTES figurent les données temporelles utilisées pour renseigner les simulations. A ce jour, le répertoire GESTES des comptes GENESIS est en général vide. Seul les utilisateurs de certaines versions ad hoc (connexion au moteur temps réel ou mode jeu embryonnaire) l'utilisent.

Le répertoire EXPORT est le répertoire par défaut proposé par GENESIS pour toutes les opérations d'exportation (sons, analyses, fichiers objets .lst).

Dans le répertoire .PHENO qui en règle générale est caché à l'utilisateur figurent l'ensemble des phénomènes (fichiers son .echf et état .stt) et analyses (fichiers .ana) relatives à un objet. GENESIS assure que l'arborescence de ce répertoire est identique à celle du répertoire STRUCTURES. Cela facilite, le cas échéant, la recherche des fichiers par les utilisateurs experts.

11.3 - L'interface du mode Lutherie

Ce dernier paragraphe constitue une introduction générale à l'interface du mode lutherie.

- Il présente les styles d'interaction, machines, périphériques et outils retenus pour concevoir GENESIS, ainsi que des éléments de la charte graphique mise en œuvre.
- Il précise les propriétés générales de l'interface, expliquant notamment pourquoi nous avons retenu l'idée d'une représentation graphique des objets CORDIS-ANIMA qui soit manipulable directement.
- Il introduit, ce faisant, la métaphore de *l'atelier de lutherie* qui organise l'ensemble des fonctionnalités.

11.3.1 - GENESIS, une interface WIMP

Styles d'interaction et Modalité

Les styles d'interaction et les modalités proposés par un système interactif conditionnent en profondeur la qualité de l'interaction avec l'utilisateur. Nos choix en la matière s'appuient sur deux types de considérations :

- L'utilisabilité : visant essentiellement à concevoir un système utilisable – ici au sens le plus large : facile à mettre en œuvre, facile à apprendre, efficace, etc. – et de par l'importance du critère de conformité, nous avons considéré qu'un certain conservatisme était préférable en matière de style d'interaction et de modalité.
- La facilité d'implémentation : pour pouvoir, dans le temps qui nous était imparti, supporter un maximum des tâches identifiées dans la partie V et avoir recours de façon poussée à la démarche *itérative* retenue pour concevoir GENESIS, il nous fallait restreindre les coûts de développement et disposer d'outils efficaces. Pour ce faire, un certain conservatisme industriel était nécessaire.

Nous avons donc décidé de faire de GENESIS essentiellement une interface de type WIMP classique.

L'utilisation de la modalité sonore appelle quelques remarques supplémentaires. On peut supposer que de nombreux utilisateurs de GENESIS ont une oreille sensible et cultivée, dont il aurait pu être intéressant de tirer profit. Nous avons considéré qu'il était au contraire essentiel de ne pas la surcharger, afin de ne pas nuire à la tâche fondamentale qu'est l'évaluation des objets par l'écoute des sons qu'ils génèrent. Nous avons donc exclu toute utilisation de la modalité sonore – autre, bien sûr, que l'écoute des sons générés.

Machine et système

Du fait de la politique du laboratoire en termes de machines de développement, GENESIS a été développé exclusivement sur machines Silicon Graphics (SGI) et pour le système d'exploitation Unix (version SGI IRIX). La portabilité du système en effet *ne figurait pas* parmi nos objectifs prioritaires.

Périphériques

Une piste tout à fait intéressante consisterait à ce que soit utilisé, au cœur de la phase de lutherie, le périphérique à retour d'effort TGR©. Il y aurait là une recherche passionnante en matière de modalité et de style d'interaction. Mais, compte tenu des

remarques portées au paragraphe 10.1 - p170 quant au degré de finalisation du système gestuel, cette piste n'est pas envisageable à ce jour. Qui plus est, le TGR© est un équipement lourd et coûteux dont les utilisateurs de GENESIS ne disposeront probablement pas à court terme hors du laboratoire.

Les périphériques retenus pour le mode lutherie de GENESIS sont donc les périphériques standards des machines Silicon Graphics :

- La souris à trois boutons.
- Le clavier, avec trois *modificateurs* (CTRL, ALT et MAJ), un pavé numérique et des touches de fonction (F1 à F12).
- Un écran couleur haute résolution.
- Un périphérique sonore – indifféremment : carte son standard ou sortie numérique – pour l'écoute des sons générés.

Outils de développement

Conformément aux remarques portées au paragraphe 3.3.2 (partie II), les outils utilisés pour développer GENESIS sont :

- La boite à outil (*toolkit*) X-Forms. Son aspect graphique et sa facilité d'emploi étaient jugées satisfaisantes dans les premières versions de GENESIS 1. Il a donc été conservé.
- La librairie graphique Open-GL pour toutes les opérations graphiques complexes.
- La librairie graphique X-Windows du système IRIX (compatible Unix) pour les gestions complexes d'évènements.
- La librairie AlAudio de SGI.
- Le langage C (norme POSIX). Nous considérons qu'un langage déclaratif comme le langage C est plus approprié lorsqu'il s'agit de développer un environnement expérimental tel que GENESIS. Développer correctement un programme à l'aide d'un langage orienté objet (C++, Java, etc.) tend à imposer une bonne connaissance préalable de l'ensemble du logiciel. Le langage C s'accorde mieux aux démarches incrémentale et itérative adoptées.
- Un environnement standard (X-Emacs, compilateur, débuggeur). Comme indiqué au paragraphe 3.3.2, nous avons préféré un environnement de développement standard à un environnement dédié aux interfaces (UIMS, etc.).

Architecture logicielle

Conformément aux remarques portées au paragraphe 3.3.4 de la partie II, l'architecture logicielle de GENESIS s'inspire à la fois des modèles MVC – quand bien même il est orienté objet – et PAC. Dans chaque agent les parties graphiques et interactives ont été autant que faire se peut séparées du noyau conceptuel. Chaque agent dispose, de plus, d'une interface pour communiquer avec les autres agents du logiciel. La communication, suivant les cas, est faite par appel de procédure, fichier, message ou mémoire partagée.

Nous ne détaillerons pas plus ici l'architecture logicielle de GENESIS. Dans le corps du document figureront, lorsqu'elles sont nécessaires, quelques précisions.

Portabilité

Il est certain que les utilisateurs terminaux de GENESIS disposent préférentiellement de systèmes Mac ou Windows. A mesure que nos travaux avançaient et qu'en parallèle la diffusion des versions successives auprès d'utilisateurs progressait, une forte demande quant à la *portabilité* du logiciel est apparue.

Force est de reconnaître que plusieurs des choix fondamentaux (notamment : le toolkit d'interface X-Forms qui est dédié au système X-Windows et l'utilisation des périphériques Unix usuels) rendent le portage vers ces systèmes malaisé, quand bien même l'architecture de GENESIS a été conçue pour le faciliter. Notons cependant qu'à ce jour :

- Un portage vers le système Linux est en cours ;
- Un portage sur un émulateur du système X-Windows pour les systèmes Mac et Windows est à l'étude. Le portage définitif de GENESIS vers ces systèmes sera effectué ultérieurement.

Elément de charte graphique

La charte graphique de GENESIS ne vise pas à conférer un *design* particulièrement original, mais de façon plus essentielle :

- A respecter le principe de cohérence – chapitre 4 paragraphe 4.5.
- A tirer parti de la notion *d'affordance* (offrande) – chapitre 4 paragraphe 4.2.
- A respecter, plus généralement, les principes et guides connus pour la communication visuelle – chapitre 4 paragraphe 4.6.

La charte graphique de GENESIS spécifie trois couleurs essentielles (bleu, vert, rouge pâle) ainsi que les conditions de leur utilisation, plus une couleur de fond et une qui connote la désactivation des *widgets*. Elle propose un ensemble d'icônes pour les actions standards. Elle fixe, enfin, les tailles standards des fenêtres.

11.3.2 - L'atelier de lutherie GENESIS

Représentation et manipulation directe : l'établi du luthier

Le choix d'une interface WIMP classique laisse une importante latitude quant au style d'interaction principal du mode lutherie, c'est à dire la façon dont l'utilisateur interagit avec l'objet d'intérêt. Deux options sont envisageables :

- La manipulation directe d'une représentation de l'objet CORDIS-ANIMA
- L'interaction langagière, un langage – compilé ou interprété – servant à spécifier l'objet[1].

Nous avons discuté au paragraphe 3.3.2 de la partie II des intérêts respectifs de ces deux styles d'interaction. Résumons les principaux points dans le cas des objets CORDIS-ANIMA.

L'interaction langagière. Elle permettrait d'effectuer aisément les actions répétitives, très souvent nécessaires lorsqu'on conçoit un réseau <MAT>/<LIA> régulier (ajout de <MAT>, connexions diverses…). Elle autoriserait l'utilisation de variables et facilite ainsi la désignation par références[2]. Elle favoriserait une pensée abstraite et conceptuelle, bien évidemment importante lorsqu'il s'agit de modélisation. Enfin, elle permettrait d'éviter les problèmes de conception inhérents à la manipulation directe – tout en induisant, il est vrai, d'autres difficultés de nature syntaxique et lexicale.

[1] Notons que c'est le choix retenu pour l'environnement MIMESIS dédié à la création graphique, dans lequel un langage compilé propriétaire est utilisé pour spécifier la *structure* de l'objet (i.e. : de son réseau <MAT>/<LIA>).

[2] L'insertion par référence d'un objet A dans un objet B implique que toute modification de l'objet A se répercute immédiatement sur chacune de ses « instances ». Dans MIMESIS, l'insertion par référence est connue sous le nom de *label*.

La manipulation directe. Une représentation graphique de l'objet CORDIS-ANIMA manipulable directement permettrait beaucoup plus aisément d'envisager l'objet comme réel et offrirait, ce faisant, un modèle mental susceptible de favoriser une démarche de modélisation physique intuitive.

Souhaitant favoriser une pensée physique intuitive de préférence à un processus de modélisation plus conceptuel[1], *nous avons très tôt décidé d'organiser le mode lutherie de GENESIS autour d'une représentation graphique et du paradigme de la manipulation directe*. Il est bien évident, cependant, que les avantages d'un tel choix sont profondément conditionnés :

- A la qualité et à l'adéquation de la représentation proposée ;
- A la qualité des interactions possibles par manipulation directe avec l'objet.

Etant donné l'importance de ces deux points, nous leur consacrons respectivement deux chapitres dans la partie suivante.

Quoi qu'il en soit, la représentation graphique des objets permet qu'une *métaphore de l'atelier du luthier* soit développée et organise ainsi l'ensemble du mode lutherie.

Dans l'atelier de lutherie GENESIS, les objets sont posés sur des *établis* Plusieurs établis peuvent être disponibles en même temps, mais à tout instant un seul est *actif* et recevant les actions de l'utilisateur. Celui-ci dispose en outre d'un *réservoir* de briques élémentaires (i.e. : de modules), d'une ou plusieurs *armoires* où sont entreposés des objets préconçus (par lui ou par d'autres), d'un ensemble *d'outils élémentaires* pour le travail des objets, d'un ensemble *d'outils plus complexes* (travail des paramètres, outils d'analyse et de mesure, etc.). Il peut en outre entreposer momentanément un objet sur des *étagères*, les gardant ainsi à portée de la main sans pour autant qu'un établi leur soit dédié.

Organisation des outils dans l'atelier ; fenêtres libres, fenêtre modales

Deux types de critères peuvent être envisagés pour classer les fenêtres qui composent l'atelier de Lutherie, suivant qu'on les envisage sous l'angle des styles d'interaction qu'elles permettent ou sous celui de leur fonction.

- Catégorisation en termes de style d'interaction

  Nous distinguons quatre styles d'interaction suivant les fenêtres.
  - La manipulation directe : dans les établis, d'abord, l'objet est manipulé directement.
  - La manipulation limitée : les objets réduits et rangés dans les étagères permettent une interaction limitée.
  - L'interaction par *widgets* : les autres fenêtres de l'interface font apparaître des *widgets*. Elles se répartissent en *fenêtres libres* et *fenêtres modales*[2].

Nous avons limité au maximum le recours à ces dernières. Seules sont modales les fenêtres qui supposent un changement radical d'objet d'intérêt et de mode de pensée et ne nécessitent pas une interaction parallèle avec les objets CORDIS-ANIMA. Tous les autres outils apparaissent dans des fenêtres *libres* que l'utilisateur peut ouvrir ou fermer en fonction de la tâche qu'il veut accomplir, et organiser à sa guise dans l'atelier[3]. Il y a là un souci de faire apparaître les outils non pas seulement comme des moyens de manipulation indirecte mais comme des compléments indispensables de la représentation centrale, qui permettent à l'utilisateur de parfaire sa connaissance de l'objet d'intérêt.

---

[1] Voir la fin de la partie II, le chapitre II de la partie III et la partie V.

[2] Rappelons : les fenêtres libres sont celles qui peuvent être ouvertes simultanément et à volonté à l'écran. Par contre, les fenêtres modales bloquent toute interaction tant qu'elles ne sont pas fermées.

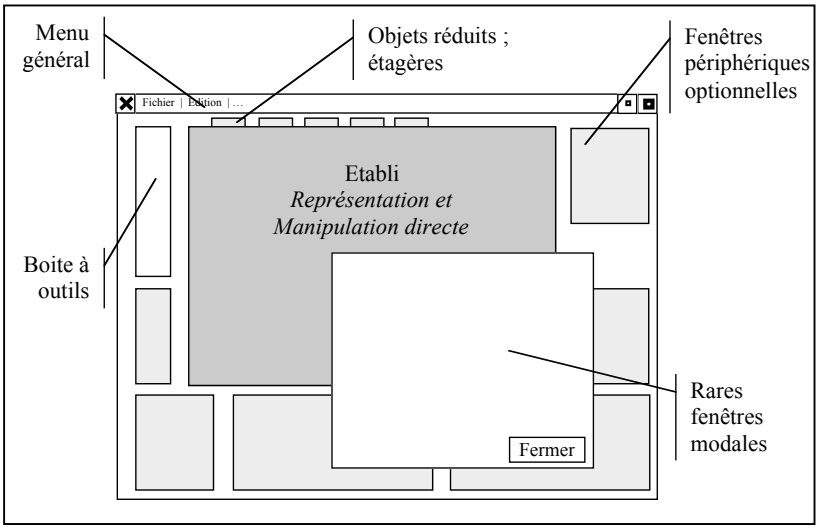
[3] On a donc une organisation proche de celle que propose le logiciel Photoshop© (Adobe).

- Catégorisation fonctionnelle

Le classement des fenêtres suivant leur fonction fait apparaître 9 entrées :

- Les établis, qui sont les lieux principaux pour l'édition des réseaux <MAT>/<LIA>.
- Les représentations réduites des objets (rangement dans des *étagères*).
- La boite à outils (ou palette).
- Le menu général.
- La fenêtre de simulation.
- Les nombreux outils d'édition de l'objet par manipulation indirecte (*widgets*).
- La fenêtre des préférences utilisateur.
- Les fenêtres d'alerte, d'erreur, de message et de choix.

Les principales fenêtres de l'Atelier ainsi que ces deux catégorisations sont résumées dans la figure16 page suivante. Une organisation typique à l'écran est proposée sur la figure 15 ci-dessous.



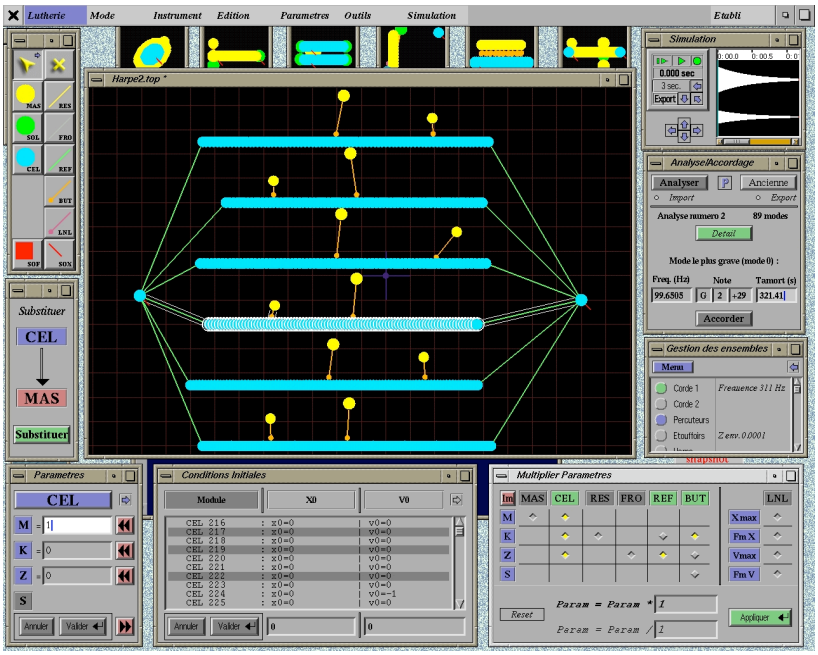


Figure 15 (couleur p. 444) : organisation générale de l'établi
Haut : catégories de fenêtre
Bas : une vue d'ensemble de GENESIS

Fenêtres 'libres'	**Etablis**	Représentation graphique des objets et manipulation directe.		Chapitre 12.
	Objet réduit ; étagère	Lorsqu'il souhaite conserver un objet à portée de la main sans plus avoir à travailler directement dessus, l'utilisateur le place dans des étagères, i.e. sous une forme réduite dans un coin de l'atelier. GENESIS propose une organisation par défaut des objets ainsi rangés, en s'appuyant sur la notion de série de structures : les objets d'une même série sont groupés. L'utilisateur peut, cependant, modifier cette organisation. Seules trois actions sont possibles sur l'objet ainsi rangé : écouter le son, reposer l'objet sur un établi (magnification), dupliquer l'objet sur un autre établi.		
	Boîte à outils	La boite à outils regroupe l'ensemble des outils permettant l'édition de la structure des objets. Elle est constamment affichée, mais peut être recouverte par l'établi si l'utilisateur est suffisamment expert pour utiliser les raccourcis.		Chapitre 13.
	Menu	Le menu du mode lutherie est unique, et placé en haut de l'écran par souci de conformité avec le système Mac. Son contenu est présenté au paragraphe suivant		Voir ci-dessous.
	Outils usuels d'information et d'édition	Plusieurs outils peuvent figurer aux côtés de l'établi, au gré des besoins de l'utilisateur. A tout instant, ces outils sont actualisés en fonction de la partie de l'objet actif à laquelle s'intéresse l'utilisateur : ils proposent, suivant les cas, un ensemble d'informations concernant cette partie et/ou un ensemble d'actions qui peuvent lui être appliqués. Ces fenêtres sont donc des fenêtres d'action indirecte.	Fenêtre de résumé	Chapitre 13.
			Fenêtre des Ensembles	Chapitre 16.
			Fenêtres des paramètres et des conditions initiales	Chapitre 14.
			Outil de génération	Chapitre 18.
			Outil de substitution	Chapitre 18.
			Outil de multiplication des paramètres (échelle)	Chapitre 18.
			Outil d'analyse modale	Chapitre 17.
	Simulation	Les fonctionnalités de simulation sont regroupées dans une fenêtre unique.		Chapitre 13.
Fenêtres modales	**Outils d'édition**	Seules deux outils d'édition apparaissent dans des fenêtres modales	Fenêtre LNL	Chapitre 14.
			Détail de l'analyse modale	Chapitre 17.
	Préférences	La fenêtre des préférences n'a pas à rester affichée ; elle peut être modale.		-
	Alertes, erreurs, etc.	L'ensemble de ces messages (qui en général nécessitent une action immédiate de l'utilisateur), sont également modaux.		

Figure 16 : catégories de fenêtres dans l'atelier de lutherie de GENESIS[1]

[1] Notons que les outils périphériques (ouverture, gestion du compte utilisateur…) ne figurent pas dans ce tableau, consacré uniquement aux outils de lutherie.

Le menu de l'atelier

Par souci de mise en conformité (ici avec les systèmes Macintosh), le menu général de GENESIS est placé en haut de l'écran.

Dans la barre des menus du mode Lutherie on trouve[1] :

- Le menu mode

Il permettra, à terme, de naviguer entre les modes Gestion, Lutherie, Jeu et Composition. Ses différentes entrées sont provisoirement désactivées dans GENESIS 1.5. Seule demeure l'entrée permettant de quitter GENESIS.

- Le menu fichier

Il est nommé ainsi par souci de mise en conformité, mais aurait pu être appelé « menu objet ». Y figurent les fonctionnalités d'ouverture, de fermeture, d'enregistrement et d'exportation d'objets, les fonctionnalité (élémentaires à ce jour) de gestion des séries, et quelques entrées permettant un accès rapide aux derniers objets édités.

- Le menu édition

Dans le menu édition figurent la fonctionnalité d'annulation[2], les fonctionnalités traditionnelles d'édition (couper, copier, coller), les fonctionnalités de gestion des Ensembles et les options d'édition pour les établis (le magnétisme par exemple).

- Le menu paramètres

Dans ce menu sont groupés les deux outils d'édition des paramètres et des conditions initiales ainsi que l'outil de multiplication (changement d'échelle d'impédance ou de fréquence).

- Le menu outils

Dans ce menu figurent les outils de génération, de substitution et d'analyse. La fenêtre des préférences est également accessible par ce menu.

- Le menu Simulation

Depuis ce menu, on peut lancer une simulation, écouter, convertir et exporter les sons, choisir enfin les options de simulation (durée par défaut des sons, par exemple).

- Le menu Etabli

Enfin, le menu établi liste les fonctionnalités de navigation (zoom, positionnement, etc.). Ce menu, en fait, vaut surtout par les raccourcis qu'il liste et permet d'apprendre. Pour naviguer, en effet, il est de loin préférable d'utiliser les raccourcis, qui ont été pensés pour être particulièrement efficaces.

---

[1] Toutes les entrées des menus appelées à être utilisées de façon récurrente sont activables par raccourcis.

[2] Par manque de temps, nous n'avons pas à ce jour développé de fonctionnalité d'annulation à plusieurs niveaux. C'est là un manque grave, qui devra être pris en compte dans les prochaines version du logiciel.

RESUME ET CONCLUSION

Cette partie a exposé les bases de GENESIS. Pour résumer, nous avons :

- Restreint la version 1.5 d'une part au temps différé, d'autre part au mode lutherie – et donné les pistes pour de futures versions du logiciel dont l'approche incrémentale de notre développement facilitera l'inclusion.
- Présenté avec précision la version de CORDIS-ANIMA que nous avons retenue, qui constitue le noyau formel au cœur de GENESIS.
- Discuté de l'organisation optimale des données, tant sur disque qu'au sein de l'interface ;
- Rapporté les choix opérés pour le développement de GENESIS : machines et systèmes, outils d'aide au développement, charte graphique, etc.
- Présenté l'organisation de l'interface du mode lutherie ;
- Introduit, ce faisant, la métaphore de l'atelier de lutherie et décidé d'organiser le logiciel autour d'une représentation graphique des objets CORDIS-ANIMA.

Dans la suite du document, nous allons détailler les principales fonctionnalités du mode lutherie.

La partie VII est consacrée aux fonctionnalités fondamentales d'édition des objets CORDIS-ANIMA.

La partie VIII introduit les fonctionnalités évoluées d'édition ; elle se partage entre l'exposé de celles que nous avons effectivement implémentées et une présentation de celles qui nous semblent à terme nécessaires compte tenu de l'analyse des tâches effectuée dans la partie V.

La partie IX, enfin, détaille les fonctionnalités et l'ergonomie mises en œuvre pour les simulations.

Partie V

Fonctionnalités fondamentales de l'atelier de lutherie GENESIS

La liste des fonctionnalités fondamentales du mode lutherie se déduit d'une analyse du processus de création en phase de lutherie. Elles doivent essentiellement permettre à l'utilisateur d'ajouter/supprimer des modules pour spécifier le réseau <MAT>/<LIA> qui constitue la structure de l'objet, de régler les paramètres et les conditions initiales de chaque module et, en support indispensable à ces actions, de prendre connaissance de l'objet à travers sa représentation.

Il est de première importance que l'ergonomie des fonctionnalités fondamentales soit particulièrement soignée. Elle détermine en effet *l'utilisabilité* de GENESIS et l'efficacité de l'utilisateur. Plus, elle conditionne la pertinence du modèle mental que se construira l'utilisateur relativement au formalisme d'une part, au fonctionnement de GENESIS d'autre part et finalement la possibilité d'un comportement intuitif face au processus de modélisation.

Nous consacrons donc l'ensemble de cette partie à l'exposé des choix effectués quant aux fonctionnalités fondamentales du mode lutherie.

Nous décrivons d'abord la *représentation graphique* des objets CORDIS-ANIMA que nous proposons (chapitre 1) puis les *manipulations directes* de cette représentation permises par le logiciel (chapitre 2).

Nous présentons les fonctionnalités relatives à *l'édition des paramètres*, qui supposent la conception de quelques objets d'interface en complément de la représentation centrale (chapitre 3).

Enfin, nous exposons dans un dernier chapitre les principales pistes que nous préconisons pour les développements futurs de GENESIS.

Contexte et bibliographie

Conception de GENESIS

Introduction

Partie I : le paradigme de la simulation interactive et multisensorielle d'objets physiques

Partie II : Interaction Homme-Machine

Partie III : Le processus de création avec GENESIS. Analyse des Tâches

Partie IV : Choix généraux pour l'environnement GENESIS

Partie V : fonctionnalités fondamentales de l'atelier de lutherie

Chapitre 12 Représentation des objets CORDIS-ANIMA 226

12.1 - ENJEUX ET PROBLÈMES 226
12.2 - PISTES 230
12.3 - RÉSUMÉ DES CHOIX OPÉRÉS 243

Chapitre 13 Manipulation directe du réseau 244

13.1 - ENJEUX ET DIFFICULTÉS 244
13.2 - PISTES ET RÉALISATIONS 246

Chapitre 14 Edition des paramètres et conditions initiales 256

14.1 - ETUDE DE CAS : EVOLUTION DE LA FENÊTRE D'ÉDITION DES PARAMÈTRES 256
14.2 - PARAMÈTRES ET CONDITIONS INITIALES DANS GENESIS 1.5 260
14.3 - EDITION DES LIAISONS NON LINÉAIRES LNL 265

Chapitre 15 Perspectives pour les fonctionnalités fondamentales 268

15.1 - ECHELLES ET VALEURS PAR DÉFAUT 268
15.2 - EXTENSION DE LA MANIPULATION DIRECTE ET DE LA REPRÉSENTATION 270
15.3 - TRANS-REPRÉSENTATION ET MULTIMODALITÉ 271

Résume et Conclusion 273

Partie VI : Fonctionnalités évoluées du mode lutherie

Partie VII : Simulation et Phénomènes

Conclusion et perspectives

Chapitre 12

REPRESENTATION DES OBJETS CORDIS-ANIMA

12.1 - Enjeux et problèmes

Tout acte de représentation consiste à mettre en vis à vis de l'objet à représenter un objet d'un nouveau type : la représentation.

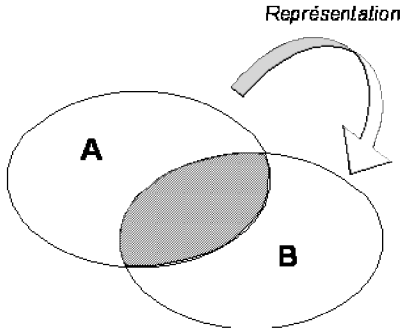


Figure 1 : Représentation
A : objet à représenter
B : représentation de l'objet
A_B : réduction de l'objet représenté
B \ A_B : apport propre
à la représentation.

Dès lors, la représentation implique[1] :

- une *réduction* par rapport à l'objet représenté ; elle ne partage avec l'objet représenté que certaines propriétés –symbolisées par l'intersection A_B sur la figure ;
- une *extension* par rapport à l'objet représenté ; des informations supplémentaires sont introduites par la représentation –symbolisées par l'ensemble B \ A_B sur la figure.

Notre objectif dans ce chapitre est de déterminer la façon optimale de porter les objets CORDIS-ANIMA à la connaissance de l'utilisateur. Pour ce dernier, l'objet qu'il conçoit s'identifie de prime abord à sa représentation. Par ailleurs, cette dernière va être le support des fonctionnalités de manipulation directe ; elle doit donc inviter à l'action, c'est-à-dire être *affordante*. La représentation, support de la perception et de l'action, est ainsi le lien essentiel entre l'utilisateur et l'objet.

D'une façon générale, la représentation doit faire ressortir les propriétés essentielles de l'objet d'une façon telle que l'on puisse les distinguer le plus aisément possible. Dans le cas des objets CORDIS-ANIMA, cependant, ces propriétés sont diverses.

12.1.1 - L'objet CORDIS-ANIMA est un réseau de modules paramétrés

En décidant de fonder GENESIS sur la maîtrise des modules élémentaires et des règles de connexion définis par CORDIS-ANIMA, nous imposons à la représentation de faire apparaître explicitement d'une part le réseau et d'autre part ses constituants que sont les modules.

[1] Voir par exemple [Cadoz.90a].

Rappelons qu'un module CORDIS-ANIMA est défini par :

- Ses points de communication. On distingue notamment les <MAT> (un point M), les <LIA> (deux points L), et les LIA à un seul point L.
- Son type (MAS, SOL, RES…) qui s'identifie d'une part à la structure de son algorithme d'autre part au comportement physique élémentaire qu'il modélise.
- Ses paramètres, qui fixent son algorithme ;
- Ses conditions initiales (dans le cas des modules <MAT>), c'est à dire la position initiale et la vitesse initiale de chacun de ses éléments matériels.

Ces quatre catégories de données sont déjà suffisamment complexes pour que la représentation d'un seul module soit problématique. Pour *chaque module* de l'objet, il nous faut établir une *hiérarchie* parmi ces données, choisir celles que nous voudrons représenter, et déterminer une stratégie de représentation.

Un autre point est lié aux caractéristiques de la version topologique de CORDIS-ANIMA utilisée dans GENESIS. L'espace de simulation d'un modèle topologique est mono-dimensionnel. Pour autant, la *complexité* du réseau n'est pas limitée. Il nous appartient donc de rendre visible à l'écran un réseau de modules d'une complexité (topologique) quelconque, alors que ses constituants se situent lors des simulations le long d'un axe unique.

12.1.2 - Echelle et inhomogénéité

La maîtrise des échelles d'un objet (et de celle des modules qui le constituent) est nécessaire en phase de lutherie[1]. Pour qu'elle soit possible, il est souhaitable que la représentation des objets permette de distinguer aisément l'échelle d'un sous-objet ou d'un module.

Par ailleurs, nous avons déjà fait remarquer qu'un utilisateur de CORDIS-ANIMA a tendance à travailler sur des ensembles de modules dont les paramètres sont homogènes. Le fait de rompre l'homogénéité d'un ensemble de module est donc une action non triviale. L'un des enjeux de la représentation est ainsi de permettre un repérage aisé des ruptures de l'homogénéité des objets.

12.1.3 - Des propriétés émergentes…

La notion d'émergence est fondamentale quand on travaille avec CORDIS-ANIMA. L'un des enjeux de la représentation est qu'elle rende compte des propriétés émergeantes du réseau de modules. Ceci signifie notamment qu'elle doit permette de distinguer :

- La structuration hiérarchique des objets, c'est à dire les différents sous-objets qui constituent l'objet complet ;
- Le cas échéant la fonction de chacun de ces sous objets ;
- De façon plus générale, certaines des propriétés des phénomènes engendrés par chacun des sous-objets, comme par exemple la fréquence, le temps d'amortissement, le timbre lorsqu'il s'agit de structures vibrantes, etc.

[1] Rappelons que l'échelle d'un objet peut être envisagée sous l'angle :
- De la valeur de ses paramètres physiques ;
- Des propriétés des phénomènes dont il est le siège (on distingue alors notamment l'échelle d'amplitude, l'échelle des fréquences et l'échelle d'impédance) ;
- Dans une moindre mesure, de la fonction de l'objet dans la hiérarchie des objets constituant le modèle complet.

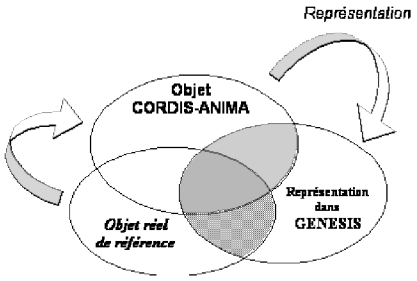


Figure 2 : Cas particulier d'émergence
la représentation devrait faire apparaître l'objet réel modélisé, quand il existe.

Enfin, évoquons le cas particulier important dans lequel l'objet CORDIS-ANIMA modélise un objet réel – corde vibrante, membrane, instrument complet, etc.

Dans ce cas, il est souhaitable que la représentation laisse entrevoir l'objet modélisé, sans pour autant que soient cachés le réseau et les modules qui le constituent (figure 2).

12.1.4 - Soutenir « l'impression de réalité » ; représentation réaliste *versus* représentation schématique

Comme nous l'avons fait remarquer dans la partie I, l'un des intérêts essentiels de CORDIS-ANIMA tient à la possibilité d'un modèle mental simple et efficace. Cette caractéristique est essentielle dès lors qu'on souhaite que l'utilisateur développe une *pensée physique*. Pour qu'elle soit mise à profit dans GENESIS, cependant, il est nécessaire qu'elle soit soutenue par la représentation des objets. Ainsi, la représentation doit tendre à ce que l'utilisateur puisse envisager les modules non pas comme des algorithmes, des concepts ou encore des modèles, mais de façon plus directe comme des objets physiques élémentaires tels des masses, des ressorts, etc.

La position diamétralement opposée pourrait cependant être adoptée.

Les modules CORDIS-ANIMA constituent des *modèles* des comportements physiques élémentaires de la matière (tel l'inertie, la viscosité, etc.). Ils ne sauraient être confondus avec leur contrepartie réelle. Plusieurs de leurs propriétés n'ont pas d'équivalent réel – et réciproquement. Cette remarque formulée pour les modules vaut également à l'échelle des objets. Par exemple, certains objets sont divergents (situation qui peut être gênante mais qui est à l'inverse parfois utilisée dans une perspective musicale). Il est quoi qu'il en soit parfaitement valable de vouloir introduire une certaine distanciation par rapport à l'idée d'un travail sur des objets réels. Les noms des modules sont, déjà, porteurs d'une telle intention (le terme « RES », par exemple, évoque le ressort mais s'en distingue). Dans GENESIS, c'est au niveau de la représentation que cette intention peut prendre corps.

La représentation est donc le lieu où se joue l'équilibre entre *réalisme* et *distanciation*. Les choix que nous avons effectués proposent un chemin entre une représentation *réaliste* (renvoyant à la réalité à travers les notions de masse ou de ressort par exemple, ou encore de corde, de membrane, etc.) et une représentation *schématique* des réseaux CORDIS-ANIMA (renvoyant à une formalisation de l'objet CORDIS-ANIMA qui apparaît alors plus comme un schéma que comme un objet).

12.1.5 - Surcharge de la représentation

En nous basant sur la Figure 1 page 226, rappelons que le choix de l'ensemble B \ A_B est de notre responsabilité.

Nous dirons qu'une information est *symbolique* ou encore qu'elle *surcharge la représentation*, dès lors qu'elle est représentée mais n'influe en rien sur les phénomènes générés. Loin de vouloir éviter toute surcharge, nous voudrons que des informations qui ne sont pas directement liées aux propriétés physiques des objets puissent être introduites dans la représentation. Pour ne donner qu'un exemple, rappelons qu'un objet CORDIS-ANIMA peut être envisagé comme une partition musicale[1], qu'il conviendrait de pouvoir commenter, ornementer, annoter, etc.

L'utilisateur est seul à même de décider des informations symboliques dont il veut surcharger la représentation des objets qu'il conçoit. Il nous appartient, par contre, de concevoir un ensemble de fonctionnalités permettant de les définir.

12.1.6 - Esthétique. Simplicité, efficacité

Nous distinguons enfin comme un enjeu à part entière *l'esthétique* de la représentation.

Nous considérons en effet qu'un utilisateur en situation de création ne peut qu'être influencé par l'esthétique du logiciel avec lequel il travaille, et particulièrement, dans le cas de GENESIS, par l'esthétique de la représentation des objets. De plus, de la même manière qu'un luthier recherche une certaine esthétique pour les instruments qu'il fabrique, l'utilisateur de GENESIS voudra que ses objets flattent la perception visuelle. Encore faut-il que cela soit possible – c'est à dire précisément, que GENESIS lui en offre les moyens.

Nous aurons recherché un certain équilibre entre simplicité et efficacité et tenté d'offrir à l'utilisateur une certaine liberté quant à la représentation.

12.1.7 - Des contraintes techniques

La librairie OpenGL est l'outil graphique que nous avons retenu. Nous disposons donc, pour faire face aux enjeux précédents, de l'ensemble des primitives graphiques traditionnelles. Nous pouvons utiliser :

- La forme des objets graphique affichés à l'écran ;
- La taille de ces objets ;
- La couleur, et éventuellement la texture des objets ;
- L'intensité de la couleur, ou sa *saturation ;*
- Un paramètre de transparence ;
- Un éventuel éclairage.

Enfin, il nous faut considérer que l'affichage du modèle est une opération récurrente alors même que l'appel des librairies graphiques est coûteux en termes de temps de calcul et risque de devenir critique pour *l'utilisabilité* du logiciel[2]. Les procédures de dessin doivent donc être particulièrement optimisées.

---

[1] Voir la partie III chapitre 2 et 3.

[2] La version 1.2 est ainsi limitée à un maximum de 500 modules environs. Dans le version 1.5 le temps d'affichage de quelques 5000 modules reste imperceptible. *L'utilisabilité* reste tout à fait correcte jusqu'à 10000 modules (sur machines Sillicon Graphics O2). Au-delà, le temps d'affichage de l'ensemble du modèle dépasse rapidement ¼ de seconde. Quoi qu'il en soit, la vitesse d'affichage n'est à ce jour jamais apparue comme étant un frein à l'utilisation de GENESIS 1.5.

12.2 - Pistes

12.2.1 - L'établi ; espace de représentation

L'espace de simulation des objets CORDIS-ANIMA topologiques utilisés dans GENESIS est monodimensionnel. Il n'est cependant pas possible de rendre compte sur un seul axe de toute la complexité d'un réseau. La représentation graphique des objets en phase de conception ne peut donc être construite sur le seul espace de simulation. Nous sommes conduit à définir un *espace de représentation* distinct.

Compte tenu de la difficulté connue à représenter à l'écran un espace tridimensionnel, et surtout à naviguer et à désigner des objets dans un tel espace, nous décidons alors d'utiliser un *plan* pour la représentation des objets.

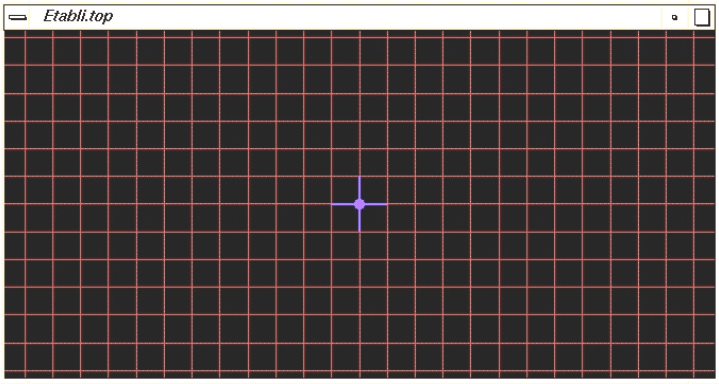


Figure 3 : l'établi du luthier, espace de représentation

L'espace de représentation défini dans GENESIS est donc le plan orthogonal à l'axe des simulations. Puisque les actions relevant de la manipulation directe y sont possibles (voir chapitre suivant), cet espace est métaphorique d'un *établi du luthier*, et le mode lutherie de GENESIS est métaphorique d'un *atelier*. Nous retiendrons le mot *établi* pour le désigner dans la suite du texte.

12.2.2 - Représentation des types des modules et du réseau.

Plusieurs versions antérieures de GENESIS ont permis d'expérimenter diverses options pour la représentation des réseaux de modules.

La version prototype de GENESIS [Cadoz&Raoult.88] propose une traduction directe du formalisme a l'écran (figure 4). Les points de communication de types M et L sont représentés explicitement. Ce choix accentue l'approche schématique de la représentation. Elle n'offre pas une *impression de réalité* satisfaisante et n'aide pas à ce qu'une *pensée physique* intuitive se développe.

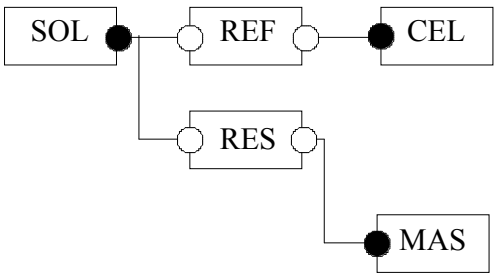


Figure 4 : la représentation proposée par la version 0 de GENESIS.

La version 1.2 [Corbun&Cadoz.95] prend une certaine distance par rapport au formalisme en ne représentant plus explicitement les points de communication. Ainsi :

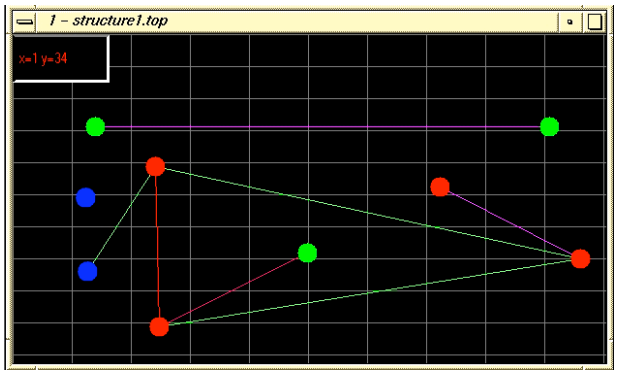


Figure 5 : la représentation proposée par la version 1 de GENESIS.

- Les MAT sont visualisés par une *pastille* circulaire placée à une position donnée de l'espace de représentation.
- Les LIA sont représentés par un segment qui lie les deux MAT connectés.

La représentation du *type* des modules (MAS, SOL, RES…) est dévolue à la couleur (figure 5). L'utilisation de la couleur est ici possible dans la mesure ou le nombre des types de modules différents est limité.

Un premier intérêt de ces choix est qu'ils sont *affordants* : ils invitent à ce que les MAT soient déplacés dans l'établi, et à ce que les LIA soient connectés aux MAT.

Par ailleurs, ils sont garants d'une certaine *impression de réalité* en laissant imaginer que les objets sont constitués de « billes » liées par des interactions. A l'inverse, le lien entre les comportements (inertiels, élastique, visqueux…) des modules et leur couleur est arbitraire. Le code de couleur implique donc une certaine distanciation, et relève de la représentation schématique.

Dans la version 2.0, le code de couleur représentant les différents types de modules est remplacé par un code graphique inspiré des représentations traditionnelles de la physique et de l'électricité. On peut raisonnablement supposer que l'utilisateur connaît les symboles traditionnels utilisés. En conséquence, la représentation de la version 2.0 est plus *explicite*. Elle permet à un utilisateur novice de comprendre la fonction de chacun des modules constituant un objet et constitue en outre une tentative vers une meilleure *impression de réalité*.

Cependant, les symboles surchargent considérablement l'écran et interdisent une bonne lisibilité de l'établi lorsque le nombre de modules s'accroît. Un symbole de petite taille, en effet, n'est pas discernable, contrairement à la couleur d'une petite surface. Les symboles imposent donc une taille minimale pour l'affichage des modules. De plus, dans la mesure où la couleur est particulièrement prégnante, il s'est avéré que *lorsque le code de couleurs est assimilé, l'association type_couleur est plus efficace que l'association type_schéma explicite.* Enfin, à nos yeux, l'utilisation de symboles est peu satisfaisant du point de vue esthétique.

Compte tenu de cette étude des expérimentations antérieures, nous avons décidé de conserver les propositions introduites par la version 1.2 de GENESIS[1], à savoir essentiellement :

- l'utilisation de *pastilles* pour la représentation des modules MAS, CEL, SOL.
- L'utilisation de *segments de droite* pour la représentation des modules de type LIA.
- L'utilisation d'un *code de couleur* pour distinguer le type du module parmi les différents MAT (ou resp. les différents LIA). Rappelons que ce code est valide puisque le nombre de types de modules est réduit. Il nécessite toutefois une *légende*, qui sera disponible dans la palette d'outils (voir chapitre suivant).

Ainsi, comme dans la version 1.2, les points M et L ne sont pas représentés explicitement. Il se peut, pourtant, qu'un LIA figure dans l'objet mais ne soit connecté à aucun MAT. Pour représenter les extrémités des LIA en instance de connexion nous avons introduit un MAT sans algorithme, *le module BLC* (blanc), qui apparaît sur l'établi comme un cercle blanc entourant le point L non connecté du LIA. Les BLC n'ont pas d'équivalent dans les simulateurs : un modèle qui en comporte ne peut être simulé. Enfin, un BLC ne peut jamais apparaître seul : il figure toujours à l'extrémité d'une ou plusieurs liaisons.

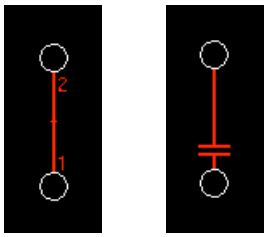


Figure 6 : propositions pour la représentation des LIA orientés

Le comportement de certains des LIA retenus pour GENESIS peut dépendre du sens donné à leur connexion par l'utilisateur. Ainsi, alors que l'orientation des LIA linéaires RES, FRO et REF est indifférente, les modules BUT et LNL sont eux des LIA orientés[2]. Il est donc nécessaire que leur représentation rende compte du sens de leur orientation. Plusieurs propositions ont été testées en ce sens (ci-contre).

La première proposition (à gauche sur la figure) s'est avérée peu satisfaisante, dans la mesure ou elle introduit un symbole alphanumérique dans une représentation qui se veut essentiellement graphique. La seconde proposition laisse supposer une liaison dont l'activité est conditionnée à la position des deux MAT, ou plus précisément encore une liaison butée. Elle est intéressante pour le module BUT, mais ne convient pas au module LNL. Elle pose en outre le même problème de surcharge graphique. Nous aurons donc opté pour une symbolisation allégée de l'orientation des BUT et des LNL. Nous la présentons dans le tableau suivant, qui résume par ailleurs l'ensemble des choix retenus pour la représentation typologique des modules.

[1] C'est principalement pour cette raison que nous n'avons pas incrémenté le numéro de version de GENESIS mais poursuivi la « version 1 », et ce même si rapidement la totalité du code originel a été remplacé.

[2] Pour le LNL c'est du moins le cas dès lors que l'une des courbes F(DeltaX) ou F(DeltaV) n'est pas symétrique par rapport à l'origine.

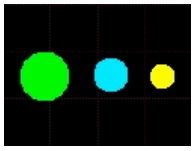
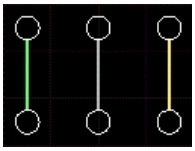
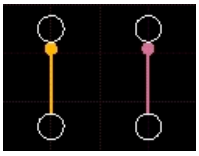
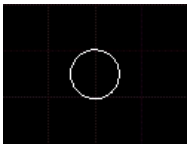
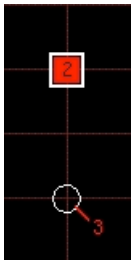
<table>
<tr><td>MAT physiques</td><td colspan="2">Les modules MAS, CEL et SOL sont représentés par une pastille circulaire. On notera qu'une telle pastille suggère aisément la notion d'élément de matière, de bille, d'inertie. La couleur est utilisée pour distinguer ces trois éléments.</td><td> 
modules
MAS, SOL et CEL</td></tr>
<tr><td>LIA linéaires physiques</td><td colspan="2">Les modules RES, FRO et REF sont représentés par un segment liant les MAT auxquels ils sont connectés. La couleur permet de distinguer ces trois éléments.</td><td> 
modules
RES, FRO et REF</td></tr>
<tr><td>LIA orientés physiques</td><td colspan="2">Les choix retenus pour la représentation de l'orientation des BUT et des LNL procède de la recherche d'une simplicité graphique. Notons qu'il n'y a pas de risque de confusion avec les MAT, d'une part du fait de la taille du disque qui marque l'orientation, d'autre part du fait de la couleur.</td><td> 
modules
BUT et LNL</td></tr>
<tr><td>LIA en instance de connexion</td><td colspan="2">L'extrémité libre d'un LIA non connecté est représenté par un cercle, symbole d'un MAT particulier : le BLC. Un objet contenant un BLC ne peut pas être simulé.</td><td> 
le module
non simulable BLC</td></tr>
<tr><td rowspan="3">Modules d'interaction</td><td>MAT</td><td colspan="2">Les modules d'interaction sont particuliers : ils marquent les points d'accès ou d'écoute.</td></tr>
<tr><td>MAT</td><td>Utilisation d'un carré pour représenter les modules SOF – et, dans les versions destinées à la recherche qui conservent des modules d'entrée, les modules ENX.</td><td rowspan="2"> 
modules
d'entrée/sortie</td></tr>
<tr><td>LIA dégénéré</td><td>Utilisation de segments de droite vers le bas de modules pour représenter les SOX.
Utilisation de segments de droite vers le haut des modules pour représenter les modules ENF.
La taille affichée des segments ne dépend pas du taux de zoom.</td></tr>
</table>

Figure 7 : représentation typologique des modules dans GENESIS 1.5 (couleur p. 444).

12.2.3 - Taille de l'établi, zoom, navigation

Dans GENESIS 1.2, la taille de l'établi s'identifie à la taille de la fenêtre en nombre de pixels, sans qu'il soit possible de zoomer ou de déplacer l'espace de représentation. Ces choix se sont rapidement avérés malvenus. Il importe qu'au contraire l'établi soit différentié de sa représentation dans la fenêtre. Un zoom est donc possible dans GENESIS 1.5[1]. Le zoom unitaire (1/1) correspond au cas ou 1 cm affiché représente 1 cm sur l'établi[2].

Il n'était absolument pas nécessaire du point de vue technique d'introduire une limite à la taille de l'établi. Cependant, il est apparu que l'utilisateur se sent « perdu » lorsqu'il est confronté à un espace virtuellement infini. L'établi de GENESIS 1.5 est donc limité : c'est une surface de 100 m de côté.

Cette dimension reste très importante. Il est essentiel que l'utilisateur ait conscience du taux de zoom courant et de la portion de l'établi effectivement visible. L'établi de GENESIS 1.5 est quadrillé par plusieurs grilles carrées de différentes tailles. Les principales qui correspondent à des taux nominaux (10,1/1, 1/10, 1/100, etc.) sont aisées à repérer. Les autres sont tout juste visibles ; elles confortent l'utilisateur dans l'idée qu'il navigue dans un espace mesuré et permettent de visualiser les pas du magnétisme.

Dans GENESIS 1.5, nous avons décidé que la *taille* des modules reste constante quel que soit le taux de zoom utilisé (figure 8). Ce choix permet d'abord que les modules (notamment les <MAT>) soient accessibles à la manipulation directe quel que soit le taux de zoom utilisé. Il autorise en outre une perception *absolue* de la taille de chaque module, propriété dont nous avons tiré parti pour la représentation des paramètres (voir ci-dessous). Il accentue enfin l'approche schématique de la représentation et contrebalance l'impression de réalité. Un tel fonctionnement s'avère opératoire. Il a cependant pour conséquence que lorsque le zoom arrière est important un sous-objet apparaît comme un amas où tous les éléments se confondent.

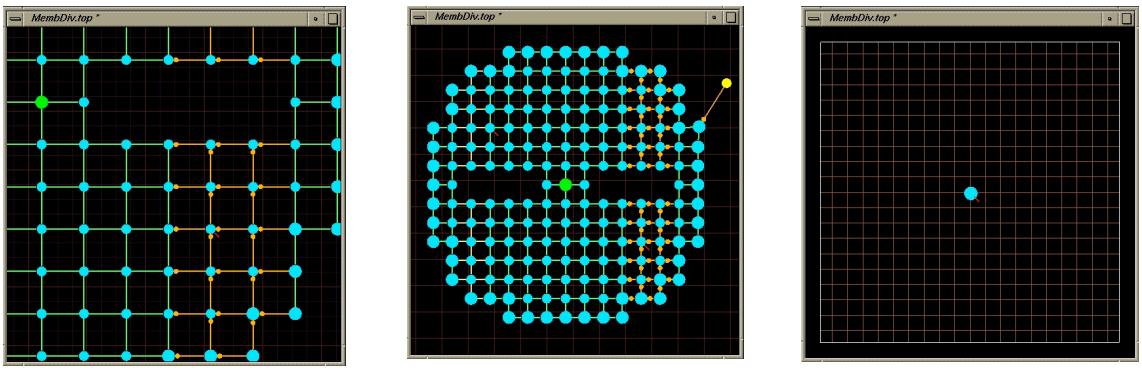


Figure 8 : le même modèle à trois taux de zoom différents. Les modules conservent la même taille apparente (Couleur p. 445)

[1] Les fonctionnalités de zoom et de positionnement – c'est à dire de navigation sont détaillées dans le chapitre suivant.

[2] Il est quelque peu abusif de mesurer les distances en cm sur l'établi, puisque dans GENESIS la plupart des dimensions n'ont pas d'unité explicite. C'est cependant un raccourci valable précisément dans la mesure ou avec un taux de zoom 1.1, un cm à l'écran correspond à 0.01 unités de distance dans l'établi.

12.2.4 - De l'utilité de l'espace de représentation

L'espace plan de l'établi et " l'impression de réalité " que dégage la représentation posent parfois un problème qu'il nous faut exposer.

La figure 9 présente trois représentations possibles d'un même objet.

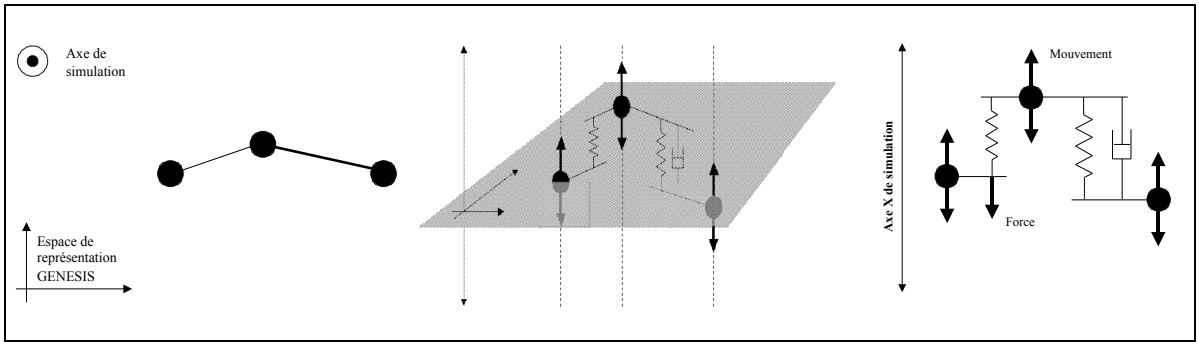


Figure 9 : trois représentations du même objet.
Gauche : représentation GENESIS
Centre : représentation fonctionnelle dans l'espace de simulation
Droite : représentation fonctionnelle.

Parmi les trois figures, celle de droite correspond le mieux au fonctionnement de GENESIS puisqu'elle ne représente que le seul axe de simulation.

Celle de gauche, qui ajoute au plan de l'établi l'axe de simulation orthogonal, est celle que nous souhaitons la plus proche du modèle mental de l'utilisateur.

Celle de droite correspond à la représentation plane de l'établi. Force est de constater que de prime abord elle risque d'induire un modèle mental problématique. Les utilisateurs novices, de fait, ont parfois tendance à conférer un sens physique au positionnement des objet sur l'établi à deux dimensions. Ils imaginent, notamment, que la distance qui sépare les MAT indique l'état de tension des LIA[1]. Pourtant, les deux dimensions de l'espace de représentation n'ont *aucun* effet sur les phénomènes générés ; elles ne constituent qu'un moyen pratique pour représenter un schéma d'interconnexions. Ainsi, alors que d'un côté on a accentué *l'impression de réalité*, de l'autre on propose un espace de représentation irréaliste.

Peter Torvik a pu interroger plusieurs utilisateurs qui se sont exprimés sur ce point. Le compositeur Giuseppe Gavazza écrit, par exemple :

> « à la première approche, j'ai eu quelques problèmes avec l'interface graphique topologique. (…) L'idée d'une masse et d'un lien élastique entre masses est très claire, mais la 'non-sensitivité' de l'élasticité à la distance séparant deux masses dans le plan de l'écran m'a étonné. (…) Pourquoi, rien ne se passe-t-il sur le ressort lorsqu'on éloigne ou rapproche sur l'écran une masse 1 et une masse 2 ? Maintenant la situation est claire et est tout à fait cohérente avec les principes fondamentaux de GENESIS, mais la représentation de la réalité à deux dimensions n'était pas immédiate et intuitive. »[2] - Giuseppe Gavazza, cité dans [Torvik.01].

---

[1] Par exemple, face à un modèle simple de corde, l'utilisateur novice imaginera qu'en déplaçant le point fixe extrême il module la tension. Il s'attendra donc à ce que la fréquence fondamentale du son généré soit modifiée par son action. Or il n'en est rien, comme on peut le comprendre lorsqu'on a intégré un modèle mental plus fidèle au fonctionnement de GENESIS.

[2] Traduction libre de l'auteur. « *on first approach I had some problem with topological graphic interface (...). The idea of mass and elasticic link between masses is very clear, but it was not immediately*

Il ne faut cependant pas exagérer la portée du problème.

D'abord, il convient de rappeler que l'introduction d'un plan de représentation est inévitable face à la version topologique de CORDIS-ANIMA.

Ensuite, il est parfaitement admissible qu'un logiciel tel que GENESIS demande quelque temps d'apprentissage et nous avons pu constater à maintes occasions que la fausse interprétation des deux dimensions de l'établi est rapidement corrigée par une courte *formation* expliquant les schémas de la figure 9. Dès lors, l'impression de réalité que dégage la représentation redevient intéressante.

Enfin, en contrepartie de l'inconvénient, une latitude fort intéressante apparaît. En effet, puisque la position des MAT dans l'espace de représentation est indifférente pour la simulation, elle constitue précisément une *information symbolique* au sens du paragraphe 12.1 -. De façon générale, comme le remarque Torvik, « le positionnement sur l'écran est libre, il n'est jamais fait au hasard par les utilisateurs sérieux » [Torvik.01]. Les deux dimensions de l'établi sont utilisées pour véhiculer une information sur les objets, pour *communiquer du sens*. Chaque utilisateur adopte une façon de travailler dans l'espace de représentation qui lui est propre – ou plutôt plusieurs stratégies suivant le contexte. Nous en décrivons à titre d'exemple quelques-unes parmi les plus courantes.

Faire ressortir la structuration hiérarchique

Les degrés de liberté offerts par l'espace de représentation sont utilisés de façon très courante pour faire ressortir la *structure* des objets complexes et surtout leur hiérarchie interne.

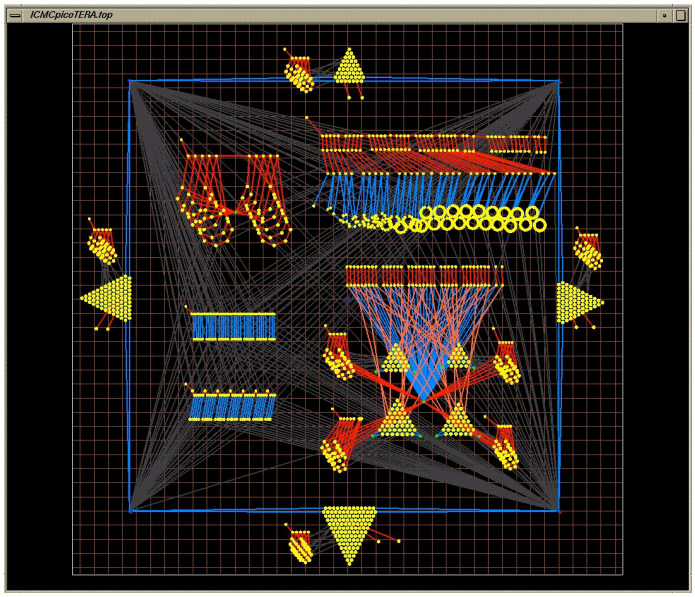


Figure 10 : aperçu général du modèle unique de la pièce Pico..TERA

- reproduit avec l'autorisation de Claude Cadoz.
(couleur p.445)

Par exemple, pour sa pièce Pico..TERA, Claude Cadoz a choisi de placer les multiples structures vibrantes à des endroits très divers de l'espace de représentation. Autour de chaque structure vibrante la dimension verticale de l'espace de représentation est utilisée pour représenter la structuration hiérarchique : les instrumentistes sont placés au dessus de la structure dont ils « jouent », les chefs au dessus des instrumentistes, et les déclencheurs à nouveau au dessus. Enfin, tout en haut de l'espace figurent les chefs et instrumentistes qui jouent sur plusieurs structures à la fois.

---

clear the « non sensitivity » of elasticity in relation with the distance between the masses on the plan of the screen. (...) Why if you put on the screen Mass1 and Mass2 closer or farer the one from the other nothing happen in ressort ? Now the situation is clear and is highly coherent with the basic principles of GENESIS, but this 2 dimensions representation of reality was not immediate and intuitive ».

Réduire l'encombrement de l'espace

Une deuxième stratégie consiste à réduire l'encombrement spatial d'une structure.

Dès lors qu'une structure linéique comprend un nombre important de MAT, il est possible de la couper en plusieurs segments sans que cela ait un quelconque effet sur les simulations, et ainsi de mieux occuper l'écran.

C'est la stratégie adoptée par exemple par Peter Torvik dans le modèle de trombone qu'il a conçu pour la pièce 11 (ci contre).

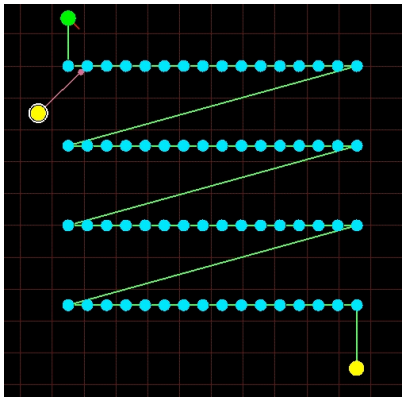


Figure 11 : un modèle de trombone conçu par Peter Torvik
(couleur p.445)

Faire ressortir certaines propriétés de l'objet, notamment paramétriques

Une autre utilisation courante de l'espace de représentation consiste à positionner les MAT de telle sorte que certaines propriétés des modules soient soulignées. Par exemple, très couramment, l'utilisateur va décaler les modules inhomogène d'un ensemble de modules pour y accéder plus aisément.

Illustrer l'objet réel modélisé

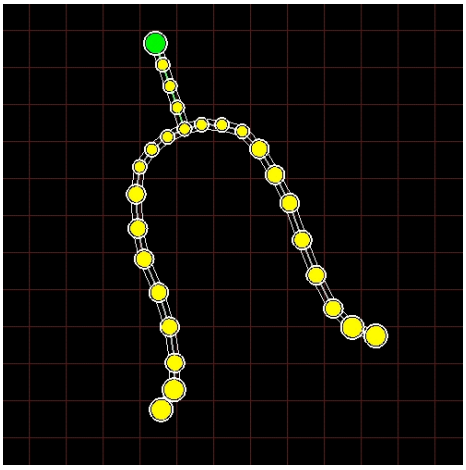


Figure 12 : un exemple de disposition évoquant l'objet modélisé
(couleur p.445)

Lorsque l'objet CORDIS-ANIMA est un modèle d'un objet réel, l'espace de représentation est souvent utilisé pour évoquer graphiquement l'objet modélisé. Cela suppose cependant que le réseau de module puisse être disposé de façon à évoquer la géométrie de l'objet réel (figure 12).

Brouiller les pistes

A plusieurs reprises, nous avons pu observer des utilisateurs utilisant l'espace pour *brouiller* la structure de leur objet, de telle sorte que précisément la représentation ne souligne aucune de ses propriétés.

Organiser le temps

L'utilisateur peur s'approprier l'une des dimensions de l'établi pour représenter, lorsque cela à un sens, le *temps.*

Par exemple, lorsque plusieurs déclencheurs sont connectés à une même structure vibrante qu'ils viennent modifier à des dates différentes, ils sont souvent placés sur l'établi dans l'ordre correspondant aux instants de déclenchement. Le même type d'option peut être choisi lorsque plusieurs structures interviennent à des moments successifs dans un modèle complexe (c'est le cas par exemple lorsqu'elles sont déclenchées successivement).

Quand GENESIS devient un outil graphique…

De façon générale, le placement des objets est travaillé avec un souci d'esthétique. Réciproquement, l'espace de représentation est parfois utilisé comme un espace *graphique*, dans lequel l'utilisateur exprime son sens esthétique puis tente de simuler le modèle obtenu. Bien qu'elle reste marginale et qu'elle s'éloigne de la " pensée physique " que nous voulons promouvoir avec GENESIS, une telle attitude témoigne à nos yeux, de l'intérêt des choix que nous avons opérés en matière d'esthétique de la représentation.

L'ensemble de ces exemples montre d'abord que si le placement des MAT n'a pas d'effet sur la simulation, il n'est indifférent ni pour le créateur ni pour celui qui découvre un modèle et doit l'analyser. On notera, ensuite, que des stratégies très diverses existent, que toutes visent à exprimer un aspect du modèle, et que ces stratégies peuvent cohabiter dans un même modèle. Il serait alors intéressant d'effectuer une étude et une classification de ces stratégies. Nous reviendrons sur ce point dans le chapitre 4 de cette partie, essentiellement prospectif, dans lequel nous nous interrogerons sur la possibilité et l'intérêt d'algorithmes de placement automatisé. Nous évoquerons de plus quelques pistes qui permettraient d'introduire une interaction réelle entre placement (par manipulation directe) et propriétés physiques des modèles – et notamment entre placement et paramètres. Nous considérons cependant que ces pistes doivent être envisagées avec précaution pour ne pas nuire aux multiples intérêts du placement libre que nous avons mis en évidence.

12.2.5 - Représentation des paramètres ; classes de paramètres

Module	Paramètres
SOL	**Aucun**
MAS	**M**
CEL	**M** **K** **Z**
RES	**K**
FRO	**Z**
REF	**K** **Z**
BUT	**K** **Z** **S**
LNL	**Deux tables :** **Force=f1(DeltaX)** **Force=f2(DeltaV)**
SOX	**Numéro de voix sonore**
SOF	**Numéro de voix sonore**

Figure 13 :
Rappel des types de paramètres

Plusieurs problèmes méritent d'être soulignés en ce qui concerne la représentation des paramètres :

- La représentation des paramètres ne doit en rien cacher le type de module, dont la perception reste primordiale.
- Le nombre de paramètres varie d'un module à l'autre (voir ci-contre). L'importance d'un module dans l'objet complet n'est cependant pas corrélée à son nombre de paramètres. Il nous faut donc éviter que la complexité de la représentation d'un module dépende par trop du nombre de ses paramètres.
- Plusieurs modules partagent les mêmes types de paramètres ; il nous faut veiller alors à ce qu'une représentation similaire soit adoptée pour les paramètres équivalents des différents modules ;

Enfin, rappelons qu'un paramètre est un caractère *numérique continu* dont la plage des valeurs est virtuellement infinie[1] et cependant pour lequel le choix entre deux valeurs proches peut être significatif. Or, si la perception humaine permet aisément de distinguer graphiquement des *catégories*, elle est bien moins efficace lorsqu'il s'agit de distinguer avec précision une valeur continue. Il n'est de fait pas envisageable que la valeur des paramètres soit perçue avec précision – sauf à l'afficher explicitement.

Les objectifs poursuivis par la représentation paramétrique doivent donc être précisés. Nous en distinguons deux :

- *Perception absolue :* nous voulons qu'une valeur approchée du paramètre puisse être perçue aisément. Pour ce faire, nous allons définir des *classes* pour chaque paramètre en nous appuyant sur une rapide étude statistique.

- *Perception différentielle :* nous voulons que l'existence d'une *différence* entre les valeurs du même paramètre de deux modules de même type soit aisément perceptible dès lors que ces modules sont affichés simultanément. De façon corollaire, nous voulons qu'un module non homogène dans une zone homogène puisse être rapidement repéré.

[1] Les notions de *continuité* et *d'infini* restant bien sûr ici conditionnées au codage numérique des nombres.

Classes des valeurs des paramètres

Une brève étude statistique des valeurs des paramètres M, K et Z de la population de modules dans un objet complexe (qui comprend notamment plusieurs échelles) mettrait en évidence une distribution à l'image de la figure suivante :

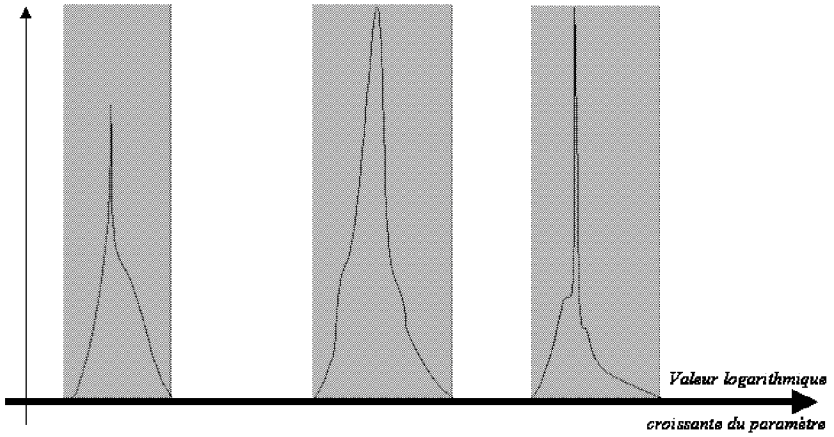


Figure 14 : répartition statistique typique des valeurs d'un paramètre M, K ou Z dans un modèle CORDIS-ANIMA

Sur cette répartition statistique typique on repère :

- L'existence de plusieurs classes – leur nombre pouvant varier d'un modèle à l'autre ;
- Pour chaque classe un point d'accumulation, autour duquel l'écart type de la distribution dans la classe est faible[1].

La stratégie de représentation des paramètres que nous adoptons s'inspire de ce profil statistique typique. Nous décidons d'expliciter au sein de GENESIS et pour chacun des paramètres M K et Z la notion de classe et de proposer une représentation pour chacune d'entre elle. Nous substituons ainsi à la représentation d'un caractère *continu* celle d'un caractère *discret*, celui de la classe à laquelle appartient le paramètre. Ce faisant, d'une part nous contournons le problème inhérent à la représentation des caractères continus, d'autre part nous précisons ce que nous entendions par perception absolue : il s'agit, en fait, que puisse être perçue la *classe* à laquelle appartient le paramètre sans qu'une comparaison entre deux classes affichées au même moment ne soit nécessaire.

Définition et représentation des classes

La répartition des classes suivant la valeur du paramètre considéré ne suit pas une loi simple – et notamment pas une loi linéaire ou logarithmique. Dans un modèle, il peut arriver que les trois classes utilisées pour l'inertie se situent autour des valeurs 1, 1000 et 1^e10. Il n'est donc pas possible d'avoir recours à une loi générale pour déterminer les classes. Au contraire, il est nécessaire que chaque classe de chaque paramètre soit explicitement déclarée dans l'interface. Dans GENESIS, la définition des classes se fait, pour chaque utilisateur, dans la fenêtre des préférences. Elle est partagée ensuite par tous les modèles.

[1] Cela laisse entendre qu'à chaque classe correspond une « valeur préférée » – propriété que nous envisagerons dans le chapitre IV.

Nous supposerons dans la suite que plusieurs classes ont été définies pour chacun des paramètres M, K et Z.

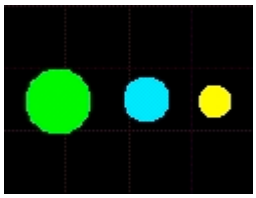


Figure 15 : trois modules MAS *dont le paramètre M appartient à trois classes différentes (couleur p. 444).*

La représentation des classes pour chaque paramètre utilise la *taille* des modules, paramètre que nous n'avions pas encore pris en compte.

A chaque classe définie pour le paramètre M correspond une taille de la pastille (voir ci contre). Les tailles peuvent être choisies par l'utilisateur dans la fenêtre des préférences. Elles ne dépendent pas du taux de zoom.

A chaque classe définie pour le paramètres K (resp. Z) correspond une *épaisseur* pour le segment le représentant. Compte tenu de la difficulté qu'il y a à percevoir de façon *absolue* l'épaisseur de segments, GENESIS choisit des épaisseurs très différentes pour chacune des classes[1].

Un problème se pose dès lors qu'un module a plusieurs paramètres. Deux solutions peuvent être adoptées et sont proposées en option dans GENESIS :

- Elire, pour chaque module, un paramètre *principal* qui puisse être représenté. Les choix suivants s'imposent alors naturellement :

CEL	REF	BUT
M	K	K

- Modifier la représentation typologique des modules de telle sorte que chacun de leurs paramètres soit représenté.

Lorsque cette seconde option est choisie, le segment unique qui représentait jusqu'ici les modules REF et BUT est remplacé par un double trait. Le premier est de la couleur des modules RES et représente le caractère élastique de la liaison. Le second, de la couleur des FRO, représente le caractère visqueux (figure 16).

Ce choix est aujourd'hui dûment testé ; il permet d'homogénéiser la représentation des paramètres K et Z sur les modules RES, FRO, REF et BUT, sans pour autant nuire à la lisibilité de l'établi.

Les CEL supposent un traitement particulier, puisqu'il s'agit alors de représenter des paramètres propres aux modules d'interaction dans un module de type <MAT>. La figure 17 présente le choix adoptés.

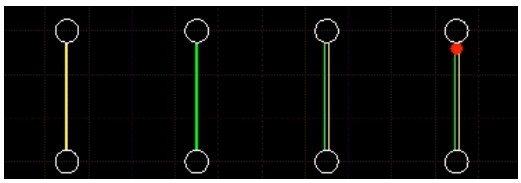


Figure 16 : représentation des modules RES, FRO, REF et BUT avec leurs paramètres *(couleur p.444).*

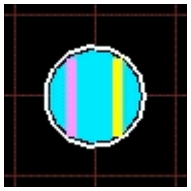


Figure 17 : représentation d'un module CEL avec ses paramètres K et Z.

[1] Un autre problème vient de ce que l'épaisseur perçue pour un segment dépend de sa couleur. Nous ne proposons pas de solution à ce problème dans GENESIS 1.5.

Enfin, il faut noter que certains paramètres ne sont toujours pas représentés : les courbes des LNL d'abord, qui sont trop complexes pour être affichées sans nuire à la clarté du schéma, les seuils des BUT ensuite, dont la signification sur l'axe de simulation peut difficilement être symbolisé dans l'espace orthogonal de représentation.

Représentation des inhomogénéités

GENESIS propose en option un deuxième niveau de représentation des paramètres, qui code non plus la classe à laquelle appartient le module mais la valeur du (des) paramètre(s) au sein de la classe. Nous utilisons pour cela la *transparence*, qui permet un effet de superposition : lorsque plusieurs MAS sont superposés du fait du taux de zoom utilisé, l'apparence de l'ensemble sera plus *intense* que l'apparence de chacun d'entre eux.

Plusieurs expérimentations nous ont convaincus qu'il serait illusoire de vouloir représenter la *valeur précise* d'un paramètre[1]. Aussi, la transparence est utilisée plus simplement pour *accentuer* les inhomogénéités dans les objets. L'algorithme mis en œuvre, dont nous proposons ci-dessous une vision synthétique, n'a pas encore été suffisamment testé pour que sa pertinence soit vérifiée. Nous considérons cependant qu'il devrait permettre un repérage aisé des éventuels modules inhomogènes dans un objet homogène.

```
Affichage
{
   Pour chaque type de module
   {
      Pour chaque paramètre
      {
         Pour chaque classe de ce paramètre
         {
```

- Déterminer le nombre de valeurs différentes que prend ce paramètre parmi les modules qui sont *effectivement visibles à l'écran* (compte tenu du zoom).
- Si N valeurs différentes sont affichées, calculer N valeurs de la transparence pour que les différentes transparences soient aussi faciles à distinguer que possible.
- Afficher les modules avec la saturation correspondante à la valeur du paramètre.

```
         }
      }
   }
}
```

Conclusion sur la représentation des paramètres

Les choix effectués quant à la représentation des paramètres couvrent une partie importante des enjeux identifiés plus haut. Une conséquence directe de ces choix, maintes fois observé, est que la notion d'échelle d'un objet – constitué de plusieurs modules – émerge aisément de la représentation.

[1] Dans la version 1.4, notamment, la transparence suivait une loi logarithmique en fonction de la valeur du paramètre dans la classe à laquelle elle se trouve. Une telle loi s'est cependant avérée inopérante. Elle ne permettait pas, notamment que les inhomogénéités soient perçues de façon satisfaisante.

12.3 - Résumé des choix opérés

Espace de représentation	Plan, orthogonal à l'axe de simulation. Placement libre des modules MAT dans l'espace. Le placement véhicule une information symbolique qui surcharge la représentation (paragraphe 1A7). Il est utilisé notamment pour : • Faire ressortir la structure et la hiérarchie de l'instrument ; • Mettre en évidence certaines propriétés de ses paramètres ; • Faire ressortir toute autre information symbolique résultat du réseau de modules ; • Véhiculer d'autres informations.
Représentation des MAT, des LIA et des connexions	Dévolue à la forme. Les MAT sont représentés par une surface (métaphore de l'élément de matière). Les LIA sont représentés par un segment (métaphore de l'élément de liaison). Les MAT connectés sont l'origine et l'extrémité du segment. Un LIA en instance de connexion est connecté à un cercle (module BLC non physique et non simulable).
Représentation typologique	Utilisation de la couleur.
Représentation des paramètres M, K, Z : échelles	Représentation catégorielle de la valeur du paramètre par la classe à laquelle il appartient. Utilisation de la taille (M), de l'épaisseur des traits (K, Z) et de la couleur (K, Z).
Représentation des inhomogénéités	Optionnelle. Utilisation de la transparence pour accentuer les inhomogénéités des paramètres des modules *visibles à l'écran*.
Seuil des butées	Non représenté.
Courbes des LNL	Non représentées.
Conditions initiales	Non représentées dans le plan de représentation.

Chapitre 13

MANIPULATION DIRECTE DU RESEAU

13.1 - Enjeux et difficultés

Les actions de *manipulation directe* sont celles qui sont opérées directement sur la *représentation de l'objet d'intérêt*, sans qu'il soit nécessaire de passer par un *objet d'interface*. Un soin particulier a été apporté à la conception des fonctionnalités correspondante, du fait des enjeux sous-jacents et des problèmes qu'elles posent.

13.1.1 - Manipulation directe et modèle mental

Puisque action et perception sont indissociables[1], le modèle mental associé à un objet d'intérêt dans une interface graphique procède à la fois de la représentation qui en est faite et des actions que permet cette dernière. Il se construit à travers une relation sensori-motrice, limitée certes puisque seule la perception visuelle est ici engagée de façon interactive, mais néanmoins fondamentale.

Un exemple simple peut être ici évoqué en ce qui concerne GENESIS. Nous avons précisé au chapitre précédent comment la représentation des MAT et des LIA laisse percevoir leur aptitude à être connectés. Cela n'est cependant pas suffisant pour que les règles de connexion définies par CORDIS-ANIMA soient assimilées de façon intuitive. Il faut encore que la *manipulation directe* permette de les pratiquer aisément, et autorise, point par point, d'effectuer les actions auxquelles la représentation invite.

Cet exemple, incidemment, peut être généralisé. Ainsi, les fonctionnalités de manipulation directe que nous concevons pour GENESIS participent, au même titre que l'aspect visuel, à l'élaboration de la représentation mentale des objets, mais aussi de CORDIS-ANIMA et plus encore de l'interface dans son ensemble. C'est là l'un de leurs enjeux essentiels.

13.1.2 - Les enjeux de la manipulation directe dans GENESIS

Puisque nous avons décidé de construire GENESIS autour d'une représentation graphique des objets, il devient essentiel que l'interactivité soit maximale et qu'un nombre aussi important que possible d'actions soient réalisables de manière intuitive par manipulation directe. Si un tel souhait rencontre tôt ou tard ses limites[2], il reste valable dans GENESIS pour plusieurs types de données qui sont du ressort de la manipulation directe, et particulièrement :

- La structure des objets, par l'ajout ou la suppression de modules et la spécification des connexions ;
- L'organisation des modules et des objets sur l'établi ;
- L'établi lui-même, à travers les fonctionnalités de navigation.

[1] Selon la théorie de Gibson – voir à ce sujet le chapitre 3.

[2] Nous n'avons pas trouvé, par exemple, de voie satisfaisante pour que les *paramètres* des objets soient manipulés directement et nous considérons que la façon la plus pertinente d'éditer les paramètres est encore de les saisir au clavier. Nous nous expliquons plus en détail sur ce point dans le chapitre suivant.

De façon générale, il est essentiel que les nombreuses actions relatives à ces données soient intuitives ou *à minima* faciles à apprendre, aisées à réaliser et efficaces.

Pour les fonctionnalités de manipulation de la structure et celles d'organisation des objets, il faut notamment que l'utilisateur puisse avoir l'impression d'agir sur des objets réels. Cela suppose entre autre qu'une majorité des actions qu'il serait susceptible de réaliser face à des objets réels aient leur contrepartie dans GENESIS.

13.1.3 - De nombreuses fonctionnalités, peu de moyens

Dans GENESIS nous avons decidé d'utiliser conjointement les trois boutons de la souris, et les trois modificateurs MAJ, CTRL et ALT du clavier. Ce sont les seuls moyens d'action de l'utilisateur, et donc les seuls dont nous disposons pour implémenter les fonctionnalités de manipulation directe. Un problème crucial qu'il nous a fallu résoudre tient donc à l'opposition entre d'une part le nombre important d'actions que nous devons permettre et d'autre part le peu de moyens techniques dont nous disposons.

Combinatoire ; utilisation des modificateurs

L'utilisation des combinaisons permises entre les modificateurs du clavier et les boutons de la souris permet de multiplier les actions possibles tout en facilitant la détermination par le programme de l'action que voulait réaliser l'utilisateur.

Le recours aux combinaisons ne doit cependant pas être systématique[1]. En effet, outre qu'il encombre la mémoire de l'utilisateur de données sans intérêt direct, il complexifie les actions élémentaires qu'il doit engager pour parvenir à ses fins et réduit l'interactivité de l'interface. Mais, le plus important en ce qui concerne les modificateurs est encore que les combinaisons utilisées *fassent système*, c'est à dire que leur organisation soit suffisamment limpide pour que l'utilisateur puisse les retenir avec un minimum d'heures de pratique[2]. C'est, entre autre, ce que nous avons recherché pour la version 1.5.

Outils

Une autre solution courante consiste à définir un ensemble d'outils. A tout instant, les actions possibles sont alors réduites à celle que permet l'outil courant. Une telle solution fait en général appel à un objet d'interface particulier : la palette graphique.

Nous avons retenu dans GENESIS le principe de *l'outil* et de la *palette*. Nous verrons cependant que plusieurs raccourcis sont proposées pour le choix de l'outil actif – de sorte que le recours à la palette n'est pas *nécessaire* pour un utilisateur averti.

Algorithmes heuristiques

Indépendamment de l'ensemble de ces stratégies, un point essentiel reste la pertinence de *l'heuristique* utilisée pour déterminer l'action que l'utilisateur veut réaliser à partir des évènements qu'il déclenche. Il s'agit que le programme « devine » ce que veut faire l'utilisateur alors même que son comportement n'est pas prévisible, et ce pour chaque outil, voire pour chaque modificateur.

L'enjeu des algorithmes heuristiques est important : eux seuls peuvent réduire le nombre d'outils et de combinaisons nécessaires à un nombre acceptable (c'est à dire notamment mémorisable), réduire le nombre de gestes élémentaires nécessaires à la

[1] C'était d'ailleurs le cas dans la version 1.2 de GENESIS, et l'une des raisons qui rendait son emploi difficile alors même qu'elle implémentait un nombre de fonctionnalité restreint.

[2] La mise en conformité du logiciel avec les pratiques courantes joue pour cela un rôle crucial, quoi qu'insuffisant.

réalisation d'une action complète et, plus important encore, rendre les fonctionnalités de manipulation directe intuitive et interactive.

Il n'est guère étonnant, dans ces conditions, que les algorithmes de décision utilisés pas GENESIS se soient complexifiés au fur et à mesure, et qu'à l'inverse les actions et combinaisons nécessaires se soient simplifiées. L'heuristique nous apparaît donc fondamentale et elle doit être encore développée.

13.2 - Pistes et réalisations

13.2.1 - Outil et palette

La palette d'outils est le premier objet d'interface de GENESIS 1.5. Les actions possibles sur la représentation sont conditionnées par l'outil sélectionné dans la palette. Cependant, la présence de la palette à l'écran n'a pas pour seule fonction la sélection d'un outil. Elle a également un impact *cognitif* et *didactique*. Elle fait office de légende et offre une représentation explicite des quelques modules élémentaires qui constituent le substrat de tout objet CORDIS-ANIMA.

Nous avons choisi d'offrir un nombre limité d'outils, chacun d'entre eux pouvant être spécialisé suivant :

- L'utilisation des modificateurs du clavier (c'est à dire la pression simultanée de la souris et de l'une des touches MAJ, ALT ou CTRL du clavier) ;
- Le bouton de la souris ;
- Et, surtout, le contexte d'utilisation, GENESIS déduisant de ce contexte la tâche que l'utilisateur souhaite effectivement réaliser. Les mécanismes impliqués de décision seront, pour partie, décrits dans la suite de ce chapitre.

Ce choix implique que certaines fonctionnalités de manipulation directe proposées par le logiciel ne sont pas explicitées dans la palette. Nous avons contourné cette difficulté :

- En limitant l'utilisation des modificateurs et des boutons de la souris (seul l'outil de navigation utilise 3 boutons), donc en développant au maximum les algorithmes contextuels qui permettent une intuition maximale.
- En nous appuyant, autant que possible, sur les habitudes supposées acquises par les utilisateurs au contact des outils informatiques courants, notamment en ce qui concerne l'utilisation des modificateurs
- En organisant le recours aux modificateurs et aux boutons de la souris de telle façon qu'un modèle mental simple et efficace soit possible.

Ergonomie de la palette

On trouve dans la palette, de bas en haut (figure 18) :

- Les outils de sélection, de modification du réseau et d'organisation à gauche (les algorithmes contextuels permettant, pour une grande part, de choisir entre ces différentes fonctionnalités) et l'outil de suppression à droite ;
- Les modules physiques du plus simple au plus complexe. Viennent ainsi : les modules linéaires élémentaires (un seul paramètre / une seule propriété physique modélisée) les modules linéaires intégrés, les modules non-linéaires. On notera que les MAT et les LIA sont positionnés respectivement à gauche et à droite.

- Enfin, les modules non physiques ou *fonctionnels*. Seuls figurent dans la palette de GENESIS 1.5 les modules de *sortie* sonore, puisque nous avons décidé de supprimer les modules d'entrée et d'interaction dans les premières versions du logiciel (voir paragraphe 1.2.7.), et d'éliminer de même dans la version GENESIS de CORDIS-ANIMA les modules de modification paramétrique et structurelle.

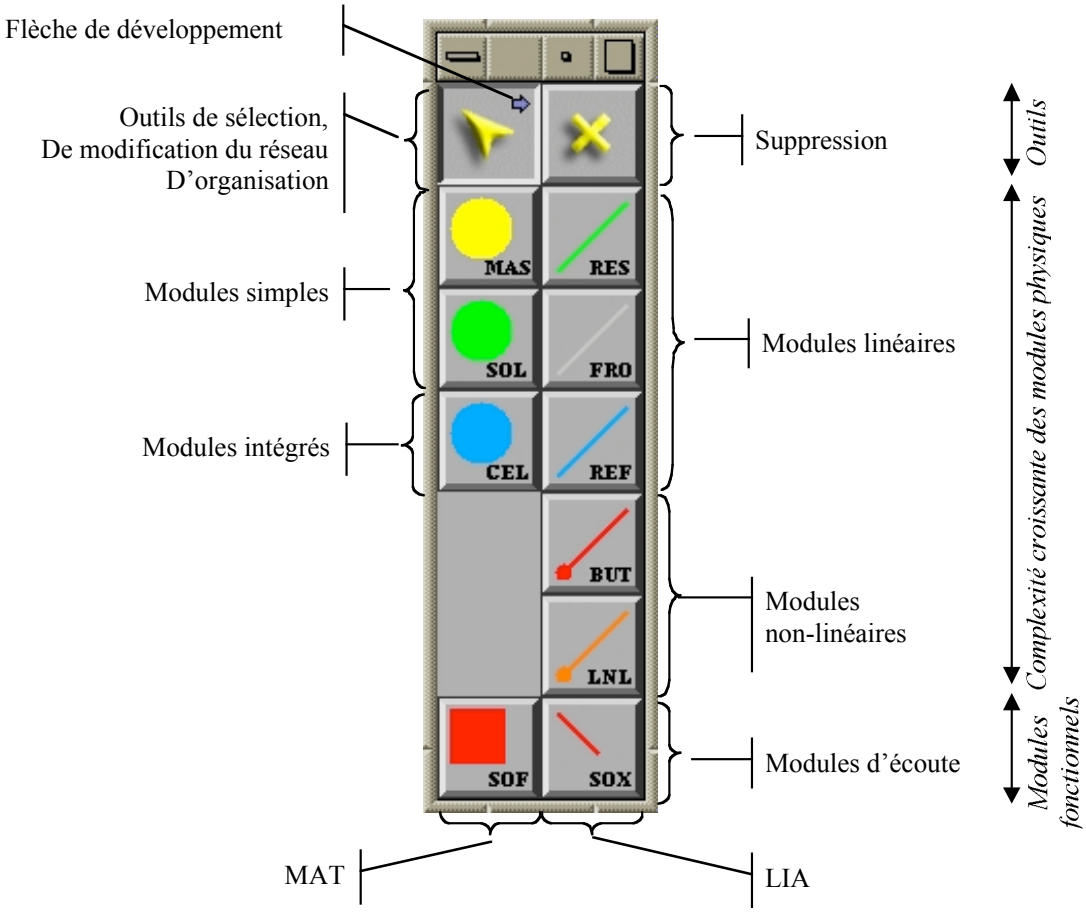


Figure 18 : La palette graphique de GENESIS *(couleur p.446)***.**

La palette utilise un principe de verrouillage. Un simple clic sur un outil permet de l'utiliser une seule fois, après quoi l'outil flèche redevient actif. Un double clic, par contre, verrouille l'outil jusqu'à ce qu'un nouvel outil soit explicitement choisi par l'utilisateur.

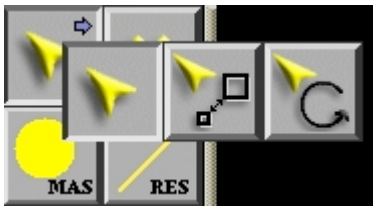


Figure 19 : le menu des outils d'organisation du réseau

A ces outils et modules immédiatement disponibles s'ajoutent plusieurs outils pour l'organisation, regroupés dans un menu (figure 19). Un algorithme heuristique élémentaire permet que ce menu ne s'affiche que lorsque l'utilisateur désire effectivement y accéder. Il peut alors y choisir :

- La *flèche*, outil essentiel permettant le déplacement des MAT et la connexion / re-connexion des LIA.
- L'outil de modification de la *taille* de la sélection ;
- L'outil de *rotation* de la sélection.

Enfin, on notera que la palette ne propose aucun outil de navigation dans l'établi (zoom, etc.). Nous avons considéré que la navigation était suffisamment essentielle et récurrente pour qu'elle doive être accessible à tout instant, sans qu'un outil n'ait à être sélectionné. Les fonctionnalités de navigation par manipulation directe seront présentées au paragraphe 13.2.7. ci dessous.

Méta-outil

Dans GENESIS, comme dans la plupart des logiciels, le modificateur CTRL est réservé aux raccourcis. Associé au clavier, il permet comme il se doit d'activer les entrées du menu. Associé à la souris, il devient un *méta-outil* qui autorise une sélection rapide des outils de la palette (figure 20).

Ainsi, un clic sur un module figurant sur l'établi avec le raccourci CTRL sélectionne l'outil d'ajout de module correspondant dans la palette. Incidemment, les paramètres du module désigné sont alors placés dans un tampon, et utilisés comme paramètre par défaut pour le module qui sera ajouté. Le méta-outil permet ainsi à moindre coût d'opérer un *copier/coller* sur un module unique.

L'activation par le méta-outil des outils de modification du réseau et d'organisation met en œuvre une interaction gestuelle plus évoluée basée sur la détection de formes implicitement dessinées lorsque l'utilisateur *tire* la souris. La détection de forme que nous utilisons reste élémentaire, du fait du peu de formes concernées, et de leur simplicité[1].

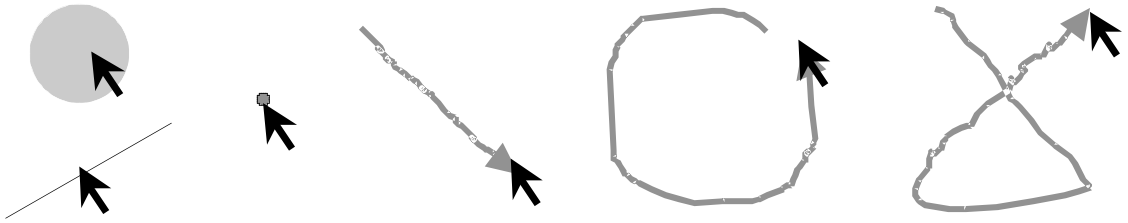


Figure 20 : Méta outil avec le modificateur ALT.
De gauche à droite, sélection rapide des outils :
Ajout de module, flèche (+ translation), rotation, taille, croix (suppression)

13.2.2 - La sélection, préalable à toute action

Dans la partie III, nous avons indiqué qu'un objet CORDIS-ANIMA est souvent constitué d'un nombre important de modules et que la lutherie suppose couramment d'agir simultanément sur plusieurs d'entre eux. En observant les utilisateurs sur les versions successives de GENESIS, nous avons effectivement constaté que l'action sur un module unique (affecter un paramètres, déplacer un MAT, connecter un LIA, etc.) est un *cas particulier*, et que le plus souvent plusieurs modules sont manipulés ensemble. Le principe de *sélection* répond à cette propriété et constitue une base pour un travail à une échelle macro-modulaire : dans GENESIS, toute action passe par la réalisation préalable d'une sélection.

[1] Nous n'avons pas utilisé, notamment, les algorithmes évolués de détection, tel celui de Rubine [Rubine.91].

Dans GENESIS 1.5 tout module peut être sélectionné de façon indépendante, de sorte que le principe de la sélection se rapproche de celui qu'implémentent les logiciels de dessin vectoriel. L'ergonomie en diffère cependant par plusieurs aspects.

Travail de la sélection

La sélection est possible avec chacun des outils de modification et d'organisation du réseau, c'est à dire aux outils flèche (translation), rotation, taille et croix (suppression).

La réalisation d'une sélection est une tâche particulièrement récurrente qu'il était essentiel de faciliter. Plusieurs stratégies sont donc proposées. La figure 22 page suivante résume les modes de travail de la sélection. Nous les commentons ci-après.

- ajouter / retrancher à la sélection

Cette fonctionnalité est dévolue de façon générale au modificateur MAJ, ce qui est parfaitement intuitif pour quiconque a l'habitude de travailler avec les logiciels courants.

- Sélection par désignation

Si le modificateur MAJ n'est pas enfoncé, un clic simple sur un module permet de le sélectionner[1] et d'enchaîner le cas échéant, en tirant la souris, l'action permise par l'outil (translation, rotation, etc…). Associé au modificateur MAJ, un clic permet d'ajouter (resp. de retrancher) le module à la sélection courante.

- Sélection par zone

Un clic sur une position libre de l'établi déclenche la fonctionnalité de sélection par zone. La seule forme disponible à ce jour pour la sélection par zone est le *rectangle*. Tous les modules figurant dans le rectangle sur l'établi ainsi que tous les LIA dont l'une des extrémités est dans le rectangle sont alors sélectionnés. Si le modificateur MAJ est utilisé, les modules sont tous ajoutés ou retranchés à la sélection en fonction de l'état de la *majorité* d'entre eux.

- Sélection par groupes de sélection

Dans la partie suivante nous introduirons la notion de *groupe de sélection,* dont la fonction essentielle est, précisément, de conserver certaines sélections particulièrement intéressantes. Nous dirons pour l'heure simplement que lorsqu'un module appartient à un tel groupe, un double clic sur le module sélectionne l'ensemble du groupe.

Représentation de la sélection

La collection des modules sélectionnés doit être visible sur l'établi sans pour autant que soit caché le réseau et les modules. Les modules sélectionnés sont, dans GENESIS, entourés en blanc (figure 21).

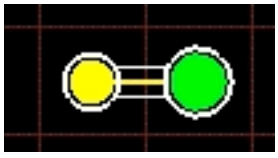


Figure 21 : trois modules sélectionnés
(couleur p.446)

[1] Notons que cela implique la définition d'une *zone de « sélectionnabilité »* autour de chaque module. Pour les MATs la zone de sélectionnabilité est une surface un peu plus grande que le module. Pour les LIA (représentés par un segment), la zone de sélectionnabilité est un rectangle allongé. Enfin, la taille de la zone de sélectionnabilité dépend de l'environnement immédiat du module. Elle grandit légèrement dès lors que le module est isolé sur l'établi. De multiples tests ont permis de valider ces choix.

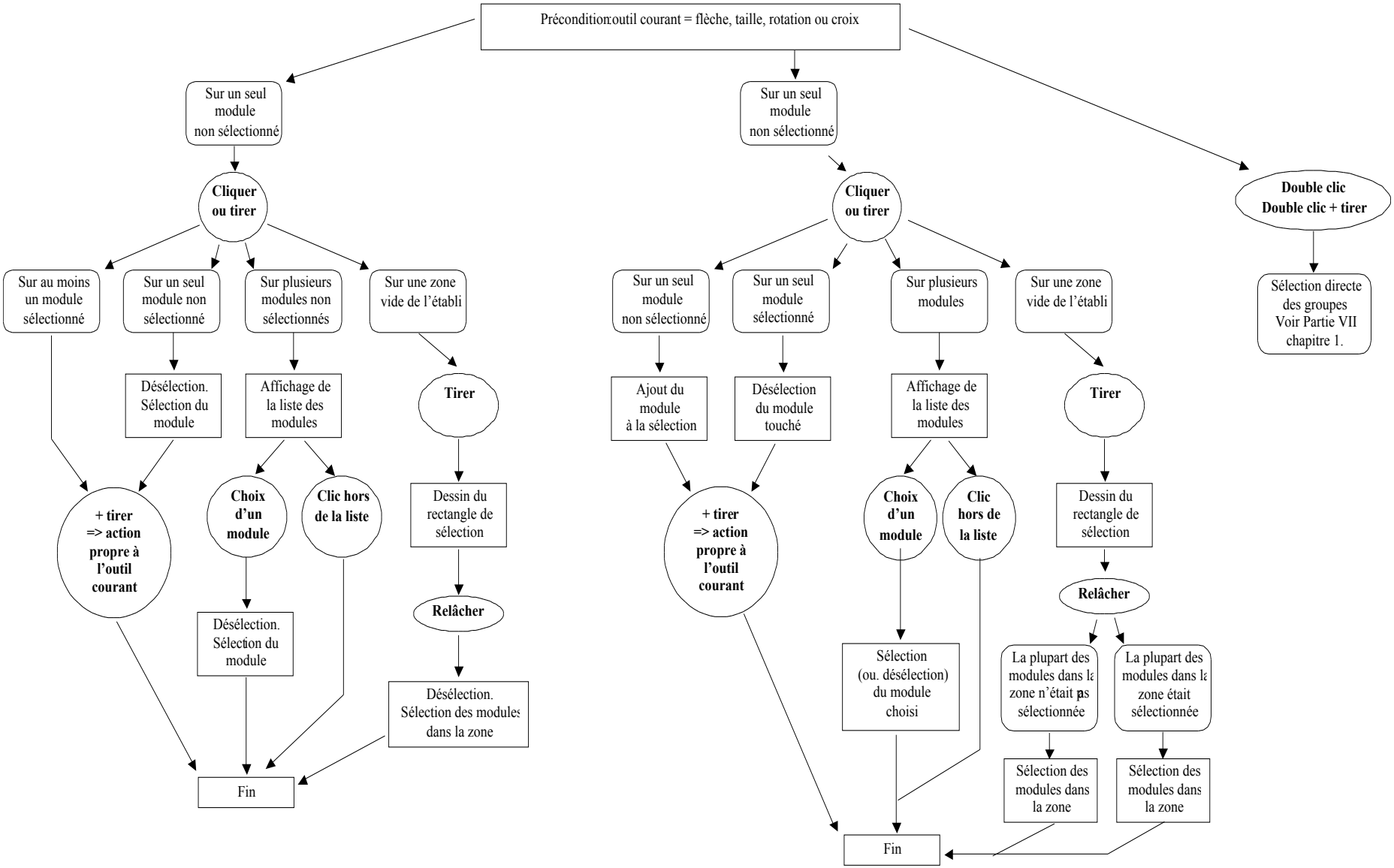


Figure 22 : Arbre de décision pour la sélection par manipulation directe (clic gauche, outil flèche, taille, rotation, croix)

13.2.3 - Ajout de modules

L'ajout sur l'établi d'un module de type donné s'effectue en sélectionnant l'outil correspondant dans la palette.

Dans les versions antérieures du logiciel, tous les paramètres des modules étaient nuls par défaut. L'observation répétée d'utilisateurs nous a permis de repérer les valeurs les plus couramment affectées aux paramètres après l'ajout d'un module. Ces valeurs sont désormais utilisées par défaut lors de l'ajout d'un module[1]. Il s'agit de :

M = 1 K = 0.01 Z = 0.0001 S=0 (pour les BUT)

Incidemment, ces valeurs assurent qu'un objet constitué pas à pas sans modification des paramètres soit une *structure vibrante*, c'est-à-dire qu'elle oscille dans le domaine audio.

L'ajout d'un module MAT est effectif dès que l'utilisateur clique sur l'établi. Si le magnétisme est activé (dans le menu édition), le module est placé sur l'un des nœuds de la grille affichée compte tenu du facteur de zoom.

L'ajout d'un module LIA fait appel à une métaphore simple du geste qui est effectué lorsqu'on raccorde deux objets par un lien, par exemple une corde : il s'effectue en *tirant* la souris entre l'origine et l'extrémité du LIA.

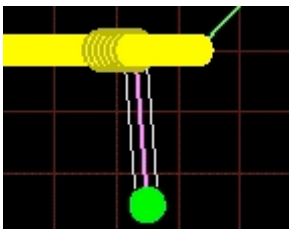


***Figure 23 : quelques modules MAT* désignés** *(couleur p.444)*.

Dès qu'un module MAT est touché, il est *désigné* par GENESIS afin que l'utilisateur sache qu'une connexion peut être établie (figure 23).

Lorsque l'utilisateur relâche la souris et qu'un MAT est touché, le LIA lui est connecté. Dans le cas contraire, un module BLC non physique est créé.

[1] Rappelons toutefois que ces valeurs par défaut valent lorsque l'outil d'ajout de module est choisi depuis la palette. Lorsque le méta outil est utilisé, les valeurs des paramètres du module ajouté sont celle du module qui avait été désigné sur l'établi.

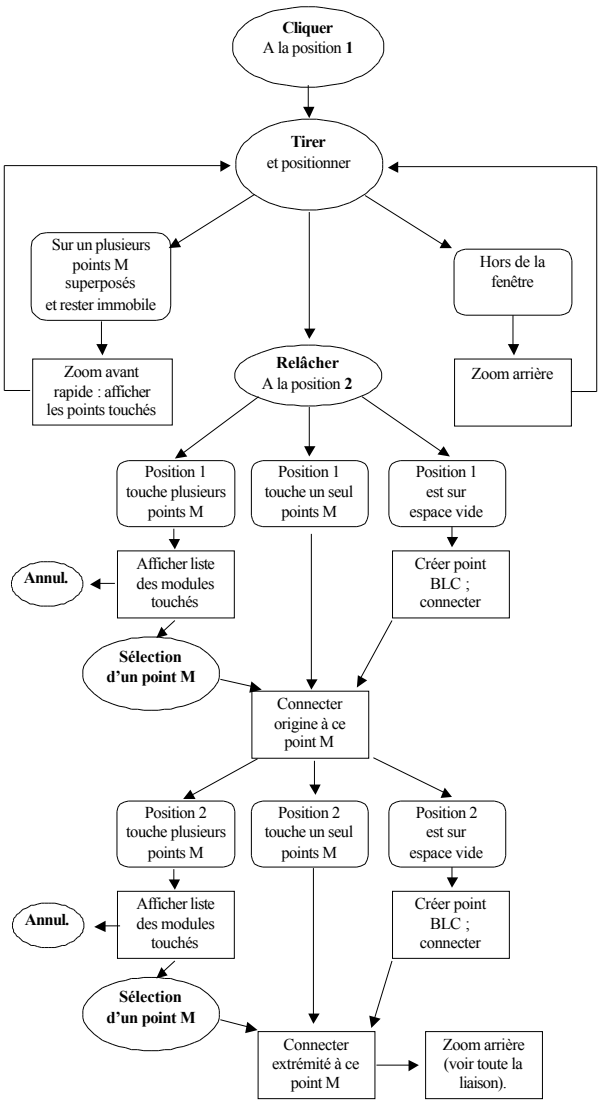


Figure 24 : Arbre de décision pour l'ajout de modules LIA

Deux problèmes se posent cependant du fait des propriétés de la représentation des objets :

- D'abord, lorsque plusieurs éléments MAT sont superposés, il faut pouvoir choisir parmi eux celui qui doit recevoir la connexion.
- Ensuite, il se peut que le module MAT auquel l'utilisateur souhaite connecter l'extrémité du LIA ne soit pas visible compte tenu du taux de zoom utilisé.

Pour répondre à ces situations, GENESIS déclenche lorsque c'est nécessaire un zoom dynamique permettant de révéler progressivement le module auquel s'intéresse l'utilisateur. La figure 24 résume le fonctionnement de l'algorithme correspondant.

13.2.4 - Déconnexion, re-connexion des <LIA>

La représentation des LIA est *affordante*: elle invite à l'action de par la forme utilisée. Une attention particulière a été accordée aux possibilités de modification des connexions du réseau MAT/LIA, de telle sorte que la pratique des règles de connexion soit à la fois facile et intuitive.

Avec l'outil flèche, l'utilisateur peut saisir l'une des extrémités d'un LIA puis la déconnecter en tirant la souris. Un module BLC est alors créé et reçoit l'extrémité du LIA devenu libre. Le BLC peut ensuite être déplacé par l'utilisateur et, si un MAT de l'objet est touché, le LIA lui sera à nouveau connecté.

Enfin, notons qu'il est possible de déconnecter plusieurs LIA en même temps, dès lors qu'ils sont sélectionnés et liés au même module MAT.

13.2.5 - Suppression de modules

La suppression de la sélection s'effectue soit à l'aide de l'outil croix (qui permet, de plus, de travailler la sélection), soit avec l'un des raccourcis claviers (CTRL+BACKSPACE) ou (SUPPR). L'ergonomie est simple et n'appelle pas plus de précision.

13.2.6 - Organisation de l'objet sur l'établi

Les outils flèche (déplacement), taille et rotation permettent à la fois de travailler la sélection et d'organiser le modèle sur l'établi. La fonctionnalité d'organisation correspondante est déclenchée dès lors que l'utilisateur tire l'un des modules MAT sélectionné – il y a là une métaphore simple de la saisie et du déplacement de la matière.

Lorsque le magnétisme est activé (dans le menu Edition), le déplacement avec l'outil flèche est fait par multiple de la taille de la grille affichée – sans pour autant que le module soit forcément placé sur l'un des nœuds de la grille.

Le modificateur MAJ permet plusieurs autres modifications sous contraintes, comme c'est le cas dans de nombreux logiciels :

- Avec l'outil flèche (déplacement), la contrainte force le module déplacé à se positionner sur la grille ; le modificateur MAJ utilisé avec l'outil flèche active ainsi un *magnétisme fort*.
- Avec l'outil taille, il force la modification à être appliquée suivant l'axe horizontal ou vertical de l'écran, ou à s'effectuer de façon proportionnelle.
- Avec l'outil rotation, il force la rotation aux multiples d'un angle élémentaire de 15 degrés.

13.2.7 - Navigation dans l'établi ; outils de navigation

Nous avons apporté une attention particulière aux fonctionnalités de navigation qui permettent à l'utilisateur de choisir, parmi le nombre important de module constituant un objet placé dans différentes zones de l'établi, ceux sur lesquels il porte son attention. Nous avons cherché d'une part à maximiser leur efficacité, d'autre part à dégager une organisation cohérente pour faciliter l'apprentissage.

Le modificateur ALT

Il n'existe pas d'outil *zoom* ou *main* dans la palette qui permette de naviguer. Nous avons en effet considéré qu'il était préférable que l'utilisateur puisse naviguer dans l'objet quel que soit l'outil sélectionné, et quelle que soit l'action en cours. Les fonctionnalités de navigation sont donc toutes accessibles avec le modificateur ALT. Associé au clavier, il permet d'accéder aisément à certains zooms particuliers. Associé à la souris, il permet de naviguer par manipulation directe.

Ce choix s'oppose au principe de visibilité puisque rien n'indique à première vue comment naviguer dans l'établi. Il garantit cependant une bonne cohérence de l'interface, de sorte que l'organisation est aisée à apprendre[1]. Il assure, de plus, que les fonctionnalités de navigation soient toujours à portée de la main et aisées à déclencher.

Taux de zoom nominaux et hiérarchie des objets

GENESIS définit six taux de zoom standards ou *nominaux* : 1000/1, 100/1… 1/1…. 1/100. Ces taux standards sont en progression logarithmique, et restent peu nombreux. Leur choix a été validé de façon empirique.

Plusieurs procédés assurent un accès préférentiel aux taux de zoom standard. L'utilisateur a donc naturellement tendance à travailler avec eux. Bien que ce ne soit pas imposé par l'interface, on peut constater qu'à chaque taux de zoom standard correspond,

[1] Les différentes combinaisons possibles sont, en outre, listées dans le menu Etabli de la barre de menus.

en général, un niveau de la hiérarchie des sous-objets. Ils apportent ainsi une aide à l'organisation de l'objet sur l'établi. Par exemple :

- Les taux 1/10 ou 1/100 évitent que les modules MAT ne soient *superposés* dans le plan de l'établi et autorisent un travail au *niveau des modules* ;
- Au taux 1/1 correspond le niveau des structures vibrantes ;
- Le taux 10/1 permet d'afficher à la fois une structure vibrante et le ou les instrumentistes qui en « jouent ». Le taux 100/1 découvre, en plus, les chefs ;
- Au taux 1000/1, enfin, les limites de l'établi deviennent visibles ainsi que l'ensemble de l'objet.

GENESIS permet un passage aisé d'un taux standard au suivant. Ainsi, l'utilisateur peut accéder sans difficulté à *toute* partie de l'établi. Il lui suffit, le cas échéant, de passer momentanément par un zoom arrière pour découvrir la portion qui l'intéresse, puis de zoomer sur cette dernière.

Aperçu

Une pression sur le modificateur ALT affiche en haut à gauche de la fenêtre de l'établi un *aperçu* (figure 25). Sa fonction est essentielle : il renseigne l'utilisateur sur la position actuelle de la fenêtre par rapport à l'ensemble de l'établi.

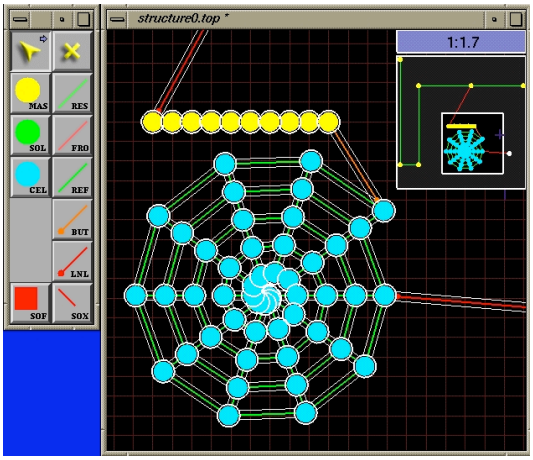


Figure 25 : aperçu (en haut à droite)
(couleur p.445)

L'aperçu est de plus accessible à la manipulation directe. En agissant sur l'aperçu l'utilisateur peut :

- Choisir un taux de zoom parmi des taux standards dans un menu ;
- Déplacer la fenêtre de l'établi par rapport au modèle en cliquant sur la zone qu'il souhaite afficher ;
- Choisir un zoom quelconque et découvrir la portion de l'établi qui l'intéresse en tirant la souris.

Navigation par manipulation directe sur la représentation

Moyennant l'activation du modificateur ALT, l'utilisateur peut naviguer dans l'établi directement à la souris.

Le bouton gauche et le bouton central permettent respectivement les zooms avant et les zooms arrière. Le zoom avant est ainsi placé sur la souris à *droite* du zoom arrière ; il y a la une métaphore peu directe mais efficace d'un *plongement*. Des pressions successives (double, triple clic, etc) permettent d'accéder aisément aux taux de zoom standard. Le bouton droit est lui dédié à l'outil *main*, traditionnel dans les logiciels de dessin.

Enfin, deux raccourcis claviers particulièrement essentiels méritent d'être cités :

- ALT + Espace, qui magnifie la sélection ;
- ALT + CTRL + ESPACE qui magnifie le modèle complet.

La figure 26 résume l'ensemble des fonctionnalités de manipulation directe accessibles pour la navigation.

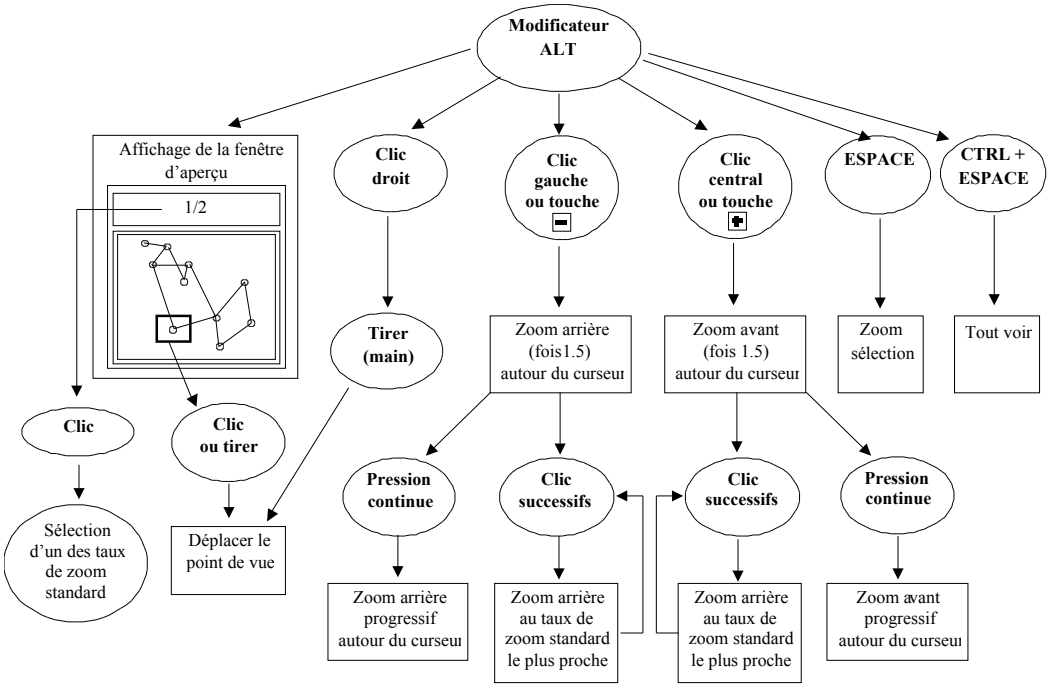


Figure 26 : navigation par manipulation directe et raccourcis

Conclusion sur la navigation

Des efforts constants auront donc été fournis pour améliorer l'ergonomie des outils de navigation ; nous considérons que d'autres devront suivre. Notamment, en nous inspirant de certaines recherches récentes sur l'ergonomie des interfaces de navigation dans les bases de données, il nous semble intéressant de mener une réflexion sur une possible interaction entre taux zoom et possibilités de manipulation directe des modules ou des ensembles de modules.

13.2.8 - Inspection : la fenêtre de résumé

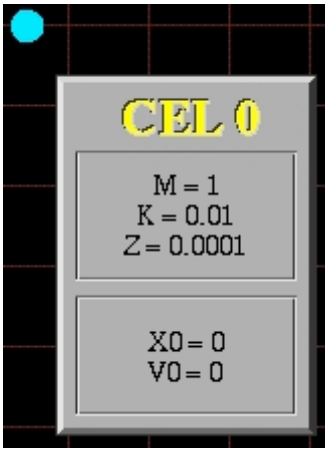


Figure 27 : la fenêtre de résumé

Nous terminerons la présentation des pistes que nous avons retenues pour la manipulation directe en évoquant la fonctionnalité *d'inspection*.

Un clic sur un module avec le bouton droit affiche la *fenêtre de résumé* (ci-contre), dans laquelle figure l'ensemble des propriétés du module touché : position dans le plan de représentation si c'est un MAT, modules connectés si c'est un LIA, paramètres, conditions initiales, mais aussi Ensembles (voir partie VI) protection (voir partie VI), etc.

L'inspection peut être conduite en tirant la souris. La fenêtre de résumé suit alors le mouvement du pointeur et affiche à tout instant les informations de l'éventuel module touché.

La fenêtre d'inspection complète avantageusement la représentation en ce qu'elle permet de prendre connaissance des informations non représentées sur l'établi, telles les valeurs précises des paramètres ou les conditions initiales. Dans le chapitre suivant, nous montrons comment ces valeurs sont éditées dans GENESIS.

Chapitre 14

EDITION DES PARAMETRES ET CONDITIONS INITIALES

Le caractère quantitatif des paramètres et des conditions initiales et, plus encore, la nécessité d'un contrôle précis impliquent de fonder leur édition sur un procédé de saisie alphanumérique[1]. Nous sommes alors contraints d'abandonner les principes de la manipulation directe et d'introduire, en vis à vis de la représentation, quelques nouveaux objets d'interfaces. Les fonctionnalités d'édition des paramètres et des conditions initiales sont cependant appelées à être utilisées très couramment : elles participent du cœur de GENESIS, au même titre que la représentation et la manipulation directe. Ce chapitre présente quelques unes des principales étapes que nous aurons franchies et résume les choix finalement adoptés.

14.1 - Etude de cas : Evolution de la fenêtre d'édition des paramètres

Au cours des trois années de développement de GENESIS, nous aurons proposé successivement six ergonomies différentes pour la fenêtre d'édition des paramètres. Quatre d'entre elles auront été implémentées. Deux, enfin, ont été validées et diffusées dans les versions 1.3 puis 1.4 de GENESIS. Une telle évolution montre que la conception d'une interface est un exercice difficile, aussi simple que soit l'objectif poursuivi ; ici la spécification des paramètres du modèle. Elle illustre la *méthodologie* que nous avons adoptée, et rend compte notamment de l'importance de l'expérience. Nous présentons succinctement dans ce paragraphe les ergonomies des versions 1.2 et 1.3. Le paragraphe suivant expose plus en détail les principes retenus finalement pour la version 1.5 de GENESIS.

14.1.1 - Edition des paramètres dans GENESIS 1.2

Présentation

La fenêtre des paramètres de GENESIS 1.2 existant aux début de nos travaux (figure 28) est modale. Elle s'affiche lorsque l'utilisateur clique avec le bouton central sur un module (édition des paramètres du module) ou sur la poignée du rectangle de sélection (édition de tous les paramètres de la sélection), et ne rend la main que lorsque l'utilisateur valide (bouton OK) ou annule les saisies effectuées.

[1] Il s'agit là des saisies de base. Nous verrons dans le chapitre V que des systèmes permettant une manipulation moins précise mais plus interactive peuvent être envisagés en complément.

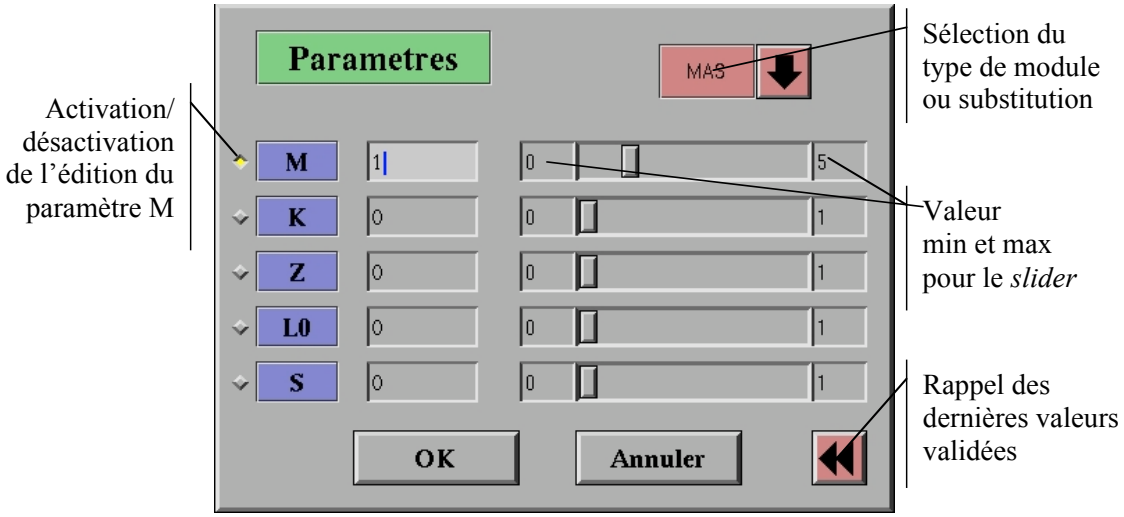


Figure 28 : fenêtre d'édition des paramètres de GENESIS 1.2

La figure 29 présente les flux d'interfaces relatifs à la fenêtre.

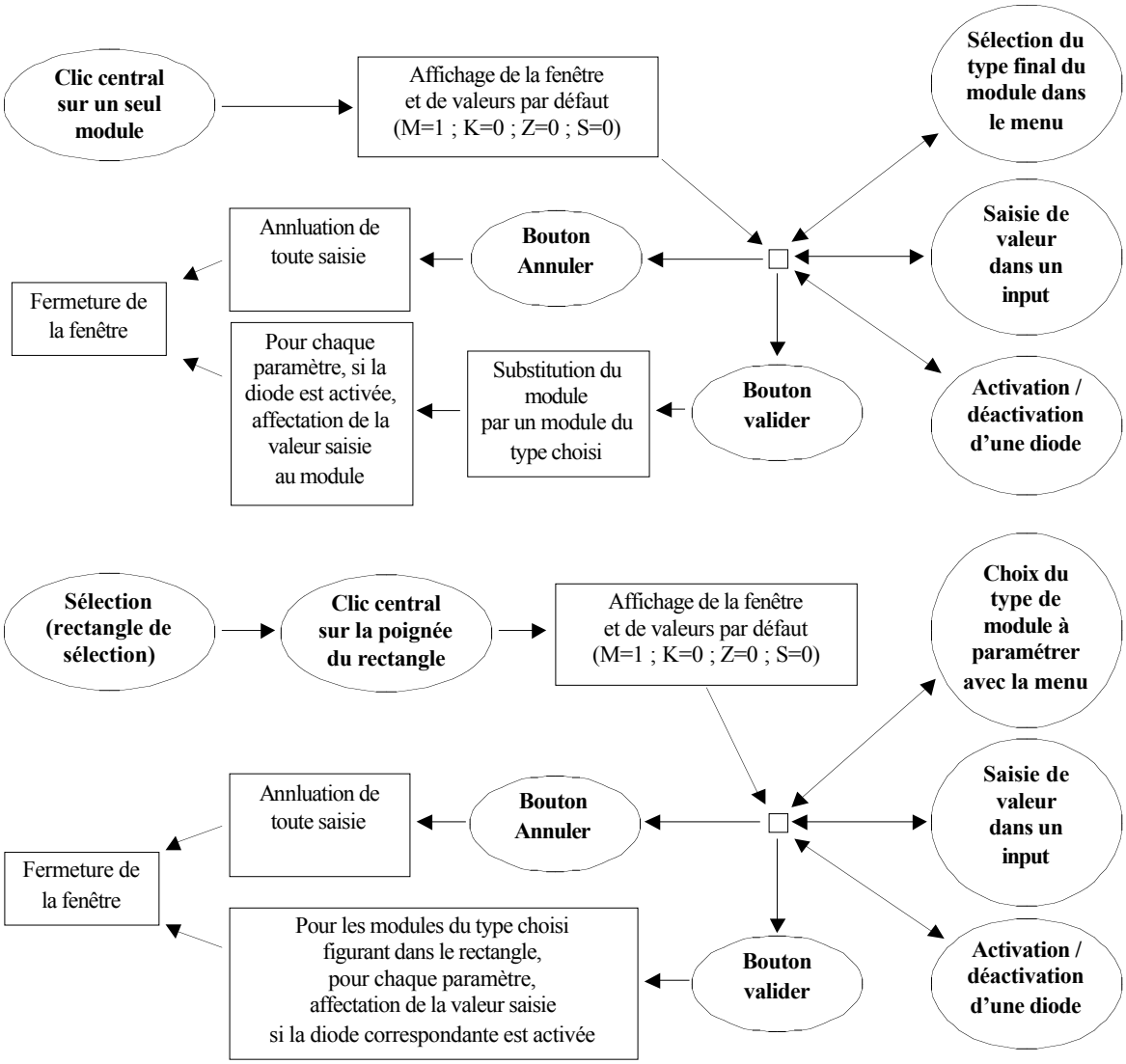


Figure 29 : flux d'interface avec la fenêtre des paramètres de GENESIS 1.2

Edition globale ; homogénéité

L'édition des paramètres des modules sélectionnés et de même type repose, dans GENESIS 1.2 sur un *principe d'homogénéité.*

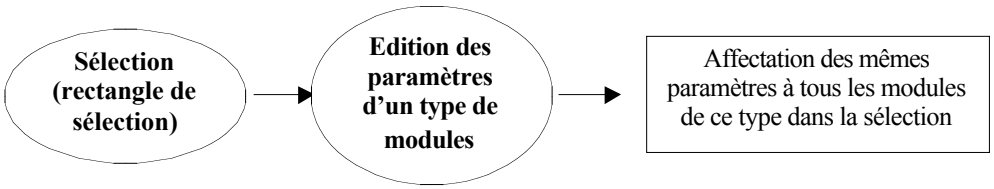


Figure 30 : principe d'homogénéité pour les paramètres

Les modèles CORDIS-ANIMA sont très souvent constitués de sous objets homogènes, ce qui est naturel compte tenu de l'homogénéité des matériaux modélisés. Le principe d'homogénéité s'avère en conséquence tout à fait pertinent. Nous l'aurons conservé dans les versions suivantes du logiciel.

Critique

D'autres propriétés de la fenêtre sont par contre malvenues. Nous en noterons quelques-unes.

- Le choix d'une fenêtre modale pour une fenêtre dont l'utilisation est récurrente est tout d'abord critiquable ; il limite considérablement *l'utilisabilité*.
- Les valeurs affichées à l'ouverture de la fenêtre sont sans rapport avec les valeurs effectives des paramètres du module ou des modules de la sélection.
- Le menu adopte un rôle très différent suivant que l'utilisateur édite un module unique ou une sélection. Dans le premier cas, il permet de *substituer* le module édité par un module du type choisi. Dans le second, il permet de choisir les modules dont on souhaite modifier les paramètres dans la sélection. Ce fonctionnement ambivalent s'oppose à l'intériorisation par l'utilisateur d'un modèle mental pertinent.
- Le principe des *sliders* est inopérant, d'une part parce que leurs valeurs minimales et maximales ne font pas sens et d'autre part, plus fondamentalement, parce que l'édition des paramètres se prête mal à une saisie approximative.

Présentation

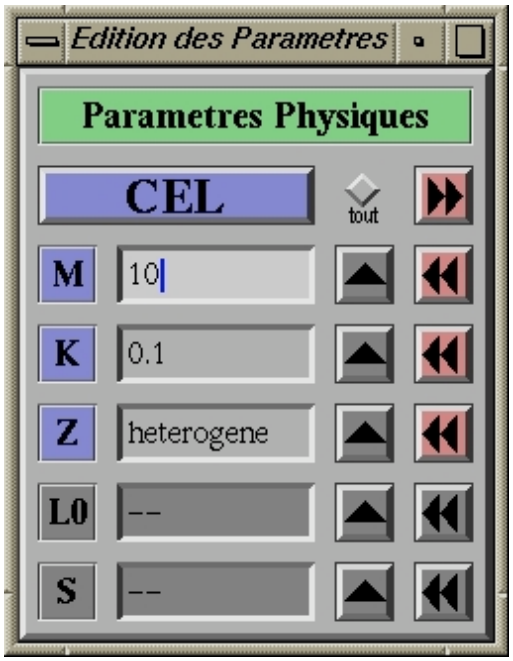


Figure 31 : fenêtre d'édition des paramètres de GENESIS 1.3

Les principales caractéristiques de la fenêtre des GENESIS 1.3 découlent des remarques précédentes.

- Elle est non bloquante. Sa petite taille – obtenue de par la suppression des *sliders* qui n'étaient pas opérationnels – permet qu'elle reste affichée sans nuire à l'organisation de l'écran.
- Elle affiche à tout instant les valeurs des paramètres de l'un des types de modules de la sélection, s'il est homogène. Le type dont les paramètres sont affichés peut être choisi soit dans le menu, soit en cliquant dans l'établi sur l'un des modules sélectionnés du type voulu. Elle acquiert ainsi une double fonction de saisie mais aussi d'exploration du modèle.
- Elle offre la possibilité de conserver les valeurs saisies pour les rappeler ultérieurement.

Flux d'interface et validation implicite

Les flux d'interface dans la fenêtre sont synthétisés sur la figure 32.

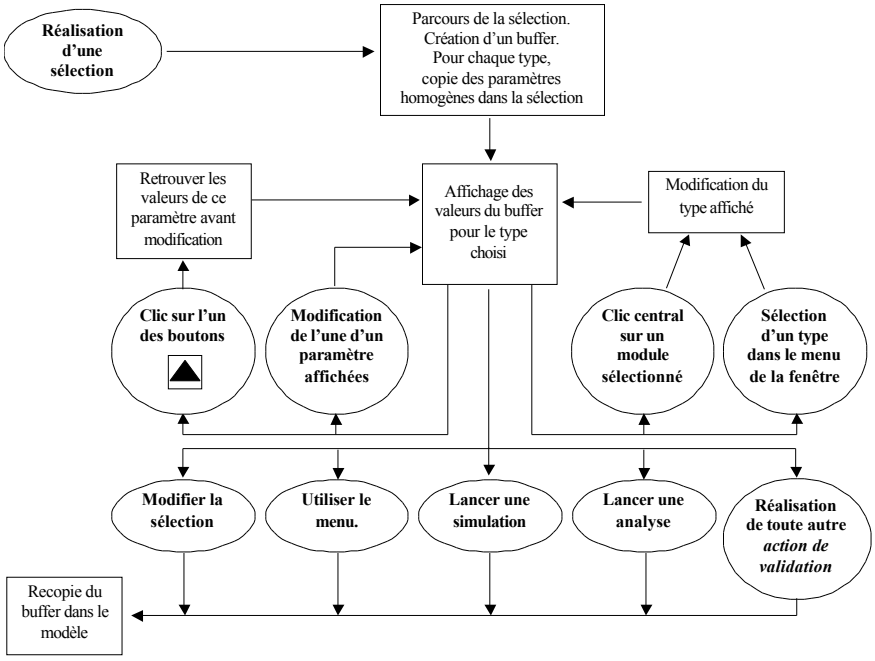


Figure 32 : flux d'interface avec la fenêtre des paramètres de GENESIS 1.3. Action de validation et validation implicite.

L'une des caractéristiques essentielles de la fenêtre est qu'aucune validation des saisies n'est nécessaire. Les saisies réalisées sont prises en compte implicitement lorsque l'utilisateur réalise une quelconque *action de validation*. *Modifier* ou *couper* la sélection, *enregistrer* ou *simuler* le modèle sont des exemples d'actions de validation.

En introduisant une telle validation implicite, nous voulions réduire le nombre des actions élémentaires nécessaires à l'affectation des paramètres. Le modèle mental qui nous paraissait sous-jacent à un tel procédé est celui d'une identité parfaite et constante entre la valeur affichée dans la fenêtre et la valeur du paramètre dans le modèle.

Malgré sa simplicité apparente, la plupart des utilisateurs de la version 1.3 ont été déstabilisés par l'absence de validation des saisies[1]. Nous avons donc reconsidéré la question lors de la conception de l'édition des paramètres dans GENESIS 1.5.

14.2 - Paramètres et conditions initiales dans GENESIS 1.5

14.2.1 - Deux types de données, deux ergonomies

L'édition des paramètres d'une part et des conditions initiales d'autre part méritent chacune une ergonomie spécifique. En effet :

- Ils représentent deux propriétés bien distinctes des modules. Les paramètres conditionnent la nature des objets alors que les conditions initiales sont des données temporelles.
- L'édition des paramètres et des conditions initiales s'effectuent à des instants différents du processus de création. En général, des conditions initiales quelconques sont d'abord utilisées pour permettre la simulation et la détermination des paramètres. Dans une deuxième phase, l'utilisateur édite les conditions initiales pour travailler l'excitation et ainsi jouer de façon élémentaire – c'est à dire organiser les évènements dans le temps.
- Enfin, alors que le principe d'homogénéité est valide pour les paramètres il est peu pertinent pour les conditions initiales.

Le choix de deux fenêtres d'ergonomie différentes, déjà présent dans la version 1.2, est en conséquence pertinent.

14.2.2 - Propriétés communes aux deux fenêtres

La figure 33 résume les flux d'interface que partagent, par souci d'homogénéité, les fenêtres des paramètres et des conditions initiales.

[1] Une critique rétrospective amène à considérer que la déstabilisation des utilisateurs provient, peut-être, de ce que le modèle mental simple n'est pas en accord avec le fonctionnement de l'interface, dans un nombre très restreint de cas, en désaccord. Par exemple, avant qu'une action de validation ne soit effectuée, un clic droit sur un module pour lesquelles des valeurs ont été saisies fait apparaître, dans la *fenêtre de résumé* les anciennes valeurs du paramètre. L'existence des boutons de rappel laissait de même entendre que les saisies n'étaient pas validées immédiatement. Il pourrait être intéressant de procéder à une nouvelle série de tests (« usability testing ») d'une interface qui validerait immédiatement chaque saisie – et qui serait ainsi en complète conformité avec le modèle mental présenté. Ce n'est cependant pas le choix qui a été retenu pour la version 1.4.

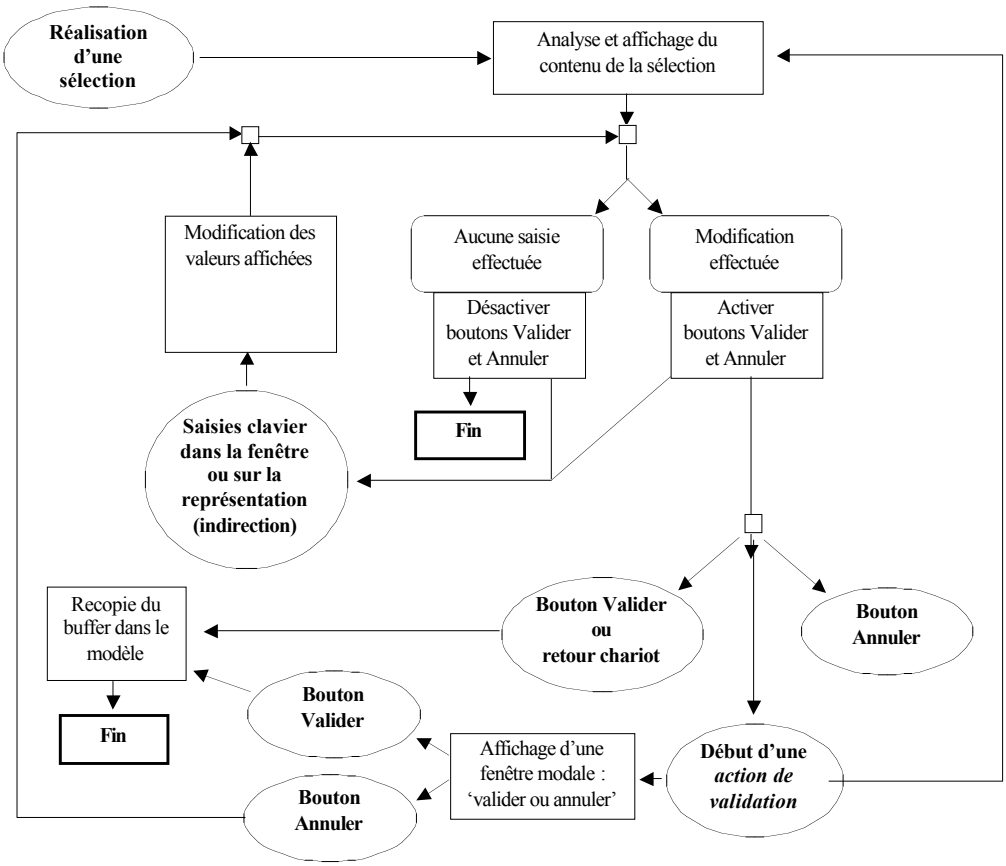


Figure 33 : Flux d'interface pour la fenêtre des paramètres et la fenêtre des conditions initiales dans GENESIS 1.5

Deux points méritent quelques commentaires.

Tout d'abord, les évènements claviers qui ont lieu dans l'établi (frappe d'un chiffre, flèches, retour chariot, etc.) sont redirigés vers la fenêtre des paramètres ou celles de conditions initiales, qui les traitent comme si elle les recevait directement[1]. Une telle *indirection* permet d'éviter les mouvements répétés de la souris entre la fenêtre d'édition et l'établi, et assurent une meilleure intégration des paramètres et conditions initiales dans la représentation centrale.

Ensuite, compte tenu des résultats des tests menés sur la version 1.3, les fenêtres d'édition de la version 1.5 nécessitent à nouveau une validation explicite des saisies.



Figure 34 : validation et annulation

Les boutons Valider et Annuler, cependant, ne sont actifs que lorsqu'une validation est effectivement nécessaire. Lorsque les valeurs saisies sont identiques aux valeurs courantes du modèle, ils sont grisés [2].

[1] Dans le cas ou les fenêtres des paramètres et des conditions initiales sont affichées simultanément, une seule d'entre elle est *active* et reçoit les évènements claviers. La fenêtre active peut être choisie à la souris (simple clic dans la fenêtre) ou depuis l'établi par un raccourci.

[2] Les boutons Valider et Annuler ici évoqués sont extraits de la *charte graphique* de GENESIS ; ils figurent dans toutes les fenêtres qui nécessitent une validation.

14.2.3 - La fenêtre des paramètres

La fenêtre des paramètres de GENESIS 1.5 est présentée sur la figure 35.

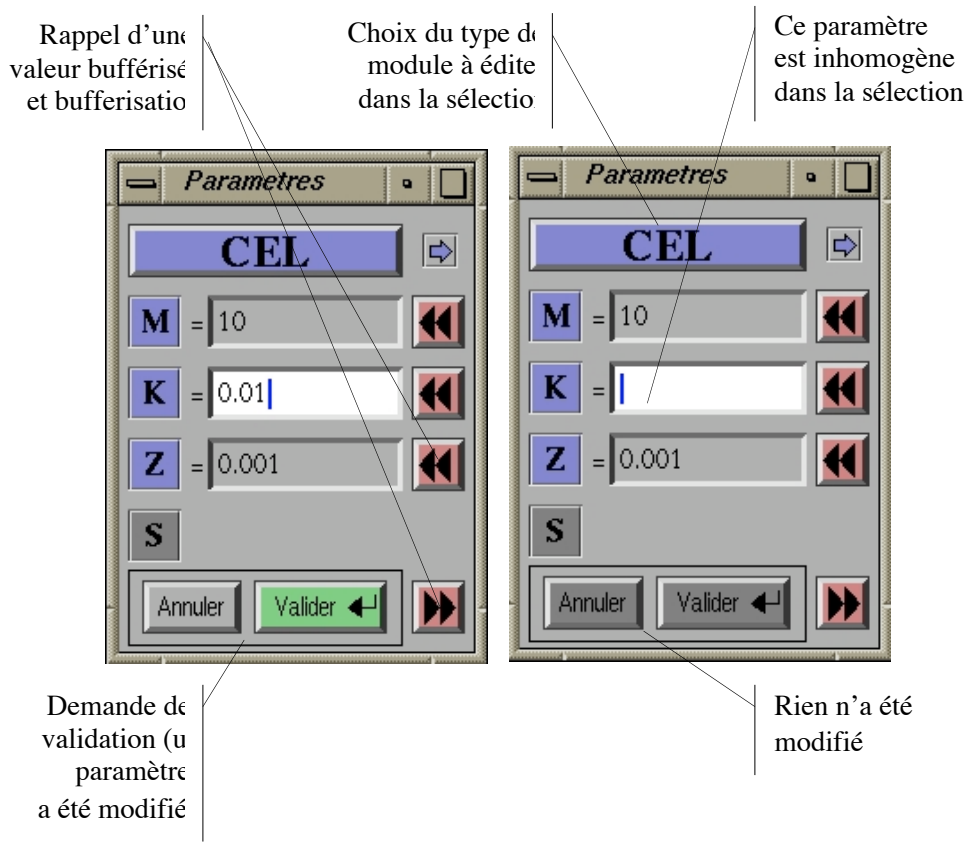


Figure 35 : fenêtre d'édition des paramètres de GENESIS 1.5 *(couleur p.446).*

On notera que les flèches de bufferisation  et de rappel 

ont été conservées. Elles permettent de diffuser sur plusieurs modules les paramètres d'un module unique. La flèche de développement  a elle pour effet d'augmenter la taille de la fenêtre, et en conséquence la précision de l'affichage.

14.2.4 - Fenêtre des conditions initiales

La fenêtre des conditions initiales (figure 36) n'est pas basée sur le principe d'homogénéité. Elle liste au contraire les conditions initiales de chacun des MAT de la sélection.

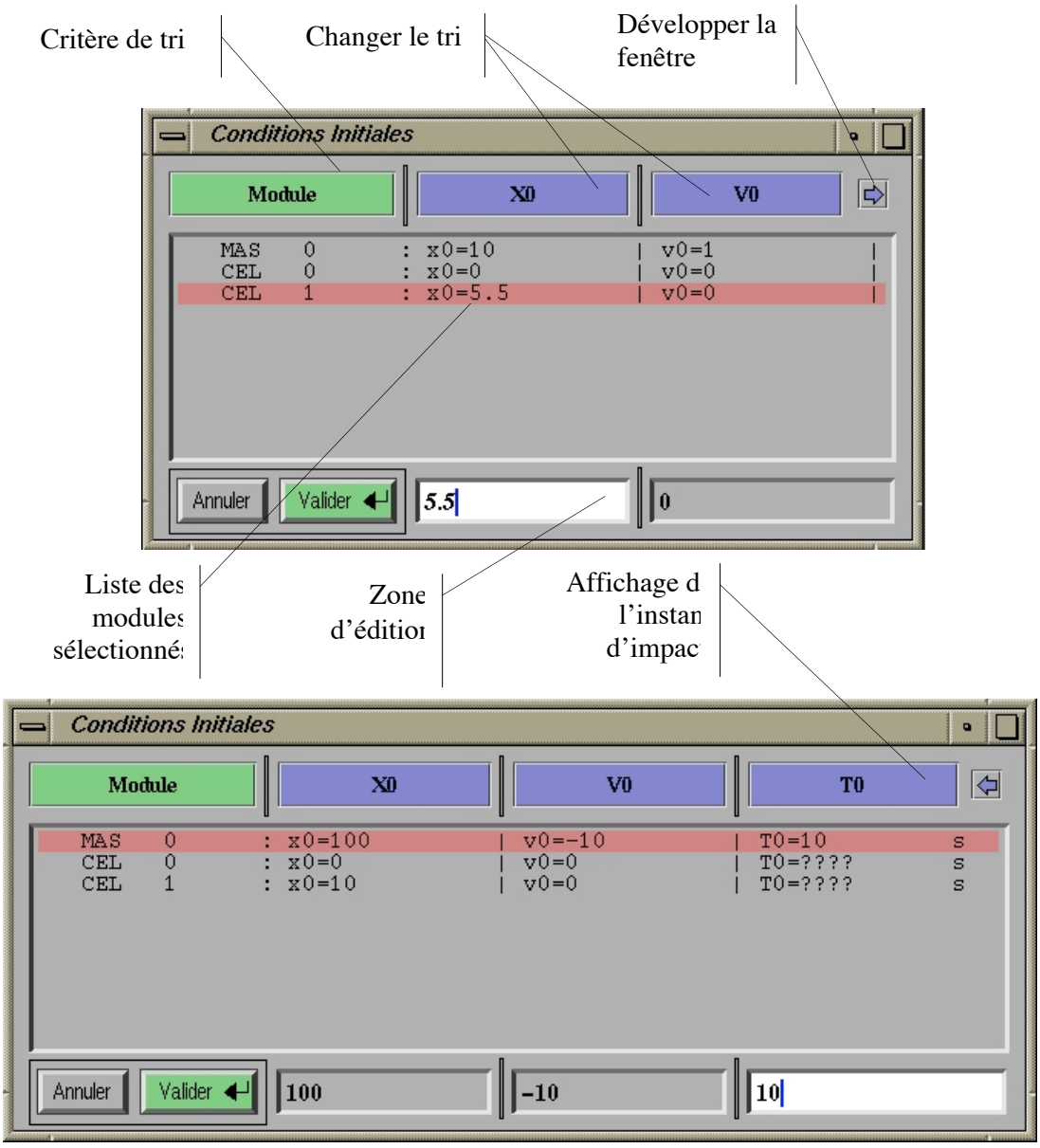


Figure 36 : fenêtre d'édition des conditions initiales de GENESIS 1.5
Haut : forme réduite. Bas : forme développée
(couleur p.446).

La saisie des conditions initiales suppose qu'un module soit choisi dans la liste. Ce module est alors surligné dans la fenêtre et *désigné* sur l'établi (voir la figure 23 ci dessus). Plusieurs modules peuvent être choisis conjointement lorsqu'une édition homogène est nécessaire (ce qui reste très rare) à l'aide du modificateur MAJ.

Le *développement* de la fenêtre mérite quelques précisions. Dans la colonne qui apparaît est alors affiché, lorsque V0 est non nul, le résultat T0 du quotient (X0/V0), c'est à dire *l'instant auquel le module couperait le plan de représentation* X=0 *si son évolution était libre*. Dans le cas ou le module MAT est un *déclencheur* (partie III 6.4), T0 est donc l'instant auquel a lieu l'impact du déclencheur sur la structure à laquelle il est connecté[1].

[1] Ceci n'est vrai en toute rigueur qu'à condition que l'objet « déclenché » est, à l'équilibre, dans le plan de représentation X=0. C'est en règle générale le cas.

T0 est un paramètre redondant avec X0 et V0. Lorsque l'utilisateur le modifie (dans *l'input* correspondant), sa valeur est répercutée sur la position initiale X0 du module, de telle sorte que *l'énergie cinétique initiale* du module (ici : du déclencheur) soit conservée.

L'affichage et l'édition de T0 permet une approche très élémentaire mais opératoire de la notion de *partition de déclencheur* (partie III paragraphe 6.5). Mais il convient de reconnaître cependant que T0 est affiché pour tous les modules même lorsqu'il n'est pas utile ou signifiant. Aussi, si sa présence a pu rendre d'importants services, il est à terme appelé à disparaître au profit d'une approche plus globale du problème[1].

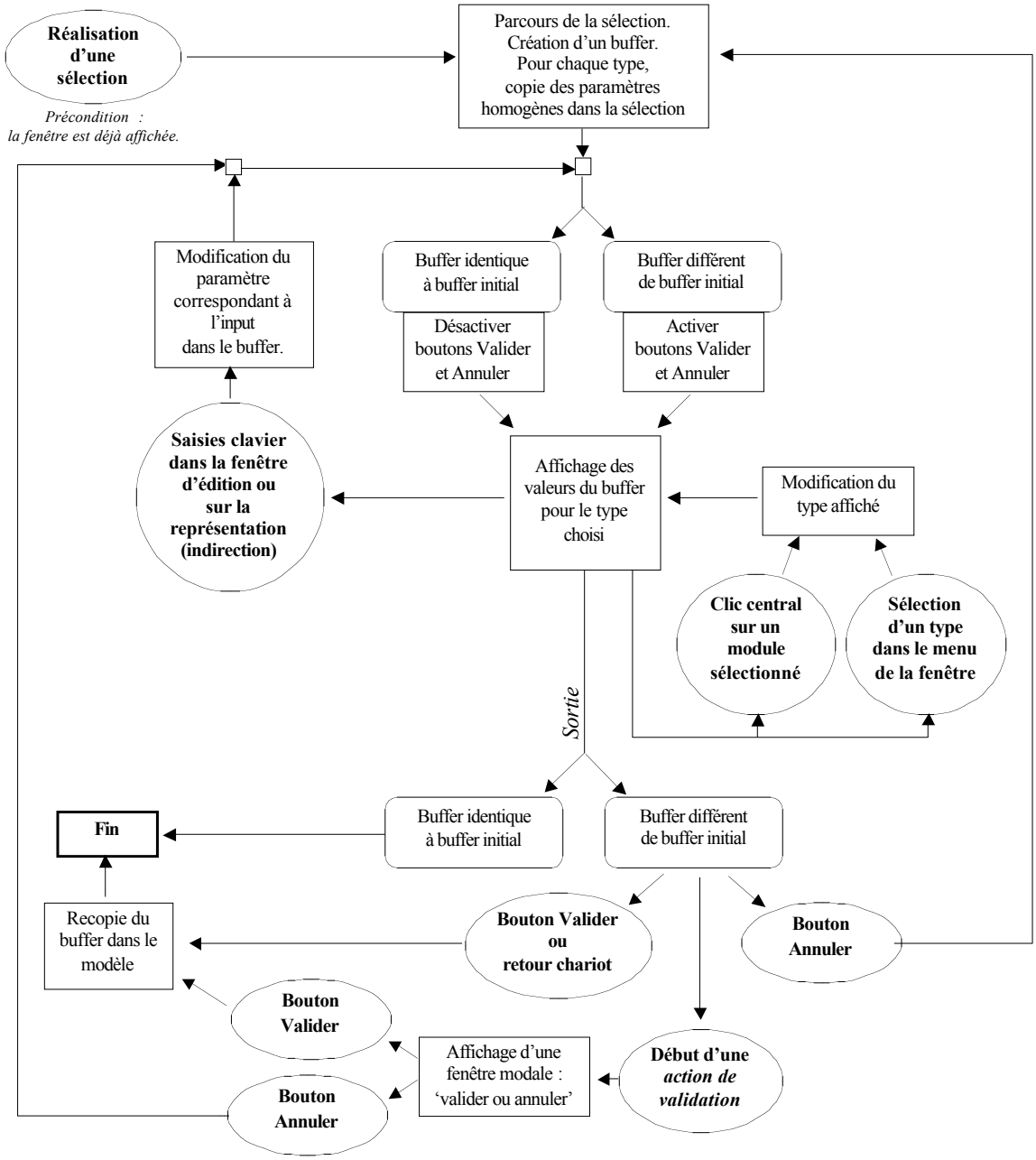


Figure 37 : flux d'interface pour la fenêtre des conditions initiale

[1] Nous présenterons dans la partie suivante le système des *relations* et *méta paramètres* encore en cours de conception.

14.3 - Edition des liaisons non linéaires LNL

Editer un module LNL, c'est spécifier deux caractéristiques force/position F_x=f(DeltaX) et force vitesse F_v=f(DeltaV), chacune de ces caractéristiques étant définie:

- par un tableau de points (ou poignées) dans le plan (DeltaX, F), resp. dans le plan (DeltaV, F) ;
- par les interpolations entre ces points : linéaire (segment de droite), par des splines ou hyperbolique.

La complexité de ces paramètres et les enjeux propres aux modules LNL nécessitent la conception d'une ergonomie spécifique. Celle que propose la version 1.5 de GENESIS résulte d'une série de prototypes et d'expérimentations qui ont permis de faire apparaître plusieurs objectifs à atteindre. Nous précisons ici les quatre principaux.

Tout d'abord, et ceci est essentiel, les LNL doivent permettre la modélisation d'une grande diversité de types d'interactions non-linéaires. Il importe que leur édition laisse un maximum de liberté à l'utilisateur.

Ensuite, dans la mesure où l'édition des LNL explicite les notions de caractéristique force/position (resp. force/vitesse), elle nécessite de la part de l'utilisateur une compréhension poussée du fonctionnement de CORDIS-ANIMA. L'ergonomie des fonctionnalités correspondante doit donc être particulièrement claire et autant que faire se peut rendre les concepts apparents.

A ces premiers objectifs s'ajoutent deux autres, plus directement liés aux types d'interaction couramment modélisés.

L'expérience montre que de nombreuses interactions non-linéaires sont *symétriques*. Cette propriété, cependant, n'est pas systématique. Il nous faut donc proposer en *option* la possibilité d'un travail sur des caractéristiques symétriques. Deux symétries sont disponibles dans GENESIS 1.5 : par rapport à l'origine ou par rapport à l'axe des forces.

Par ailleurs, il s'avère qu'il est souvent intéressant de pouvoir modifier *l'amplitude* des caractéristiques sans rien changer à leur *forme*. C'est le cas, par exemple, lorsque le LNL modélise une interaction archet/corde, un pincement, etc. De fait, l'édition des LNL fait en général apparaître deux phases. Pendant la première, l'utilisateur travaille la forme de la caractéristique, qui est représentative du type d'interaction souhaitée. Pendant la seconde, il modifie son amplitude. Une modification de l'amplitude suivant l'axe des abscisses DeltaX (resp. DeltaV) affecte, d'une certaine manière, la sensibilité du LNL. Une modification suivant l'axe des forces affecte l'importance des effets produits dans l'objet. Quoi qu'il en soit, il est souhaitable que l'interface permette un contrôle *indépendant* des formes et des amplitudes des caractéristiques.

De façon plus générale, nous avons vu que les paramètres des LNL ne sont pas représentés sur l'établi. C'est donc au niveau des fonctionnalités d'édition qu'il nous faut permettre à l'utilisateur de prendre connaissance des deux caractéristiques de la façon la plus immédiate possible.

Les deux derniers points nous ont conduits à concevoir une ergonomie à deux niveaux. Le premier propose à une *vue d'ensemble* des caractéristiques du module édité alors que le second est dédié à leur *édition profonde* et offre un maximum de précision.

14.3.1 - Les LNL dans la fenêtre des paramètres : visualisation des caractéristiques et modification d'amplitude

Le premier niveau est intégré à la fenêtre des paramètres. Il permet :

- de visualiser la forme de chacune des deux caractéristiques
- de *modifier* leur amplitude suivant chacun des axes.

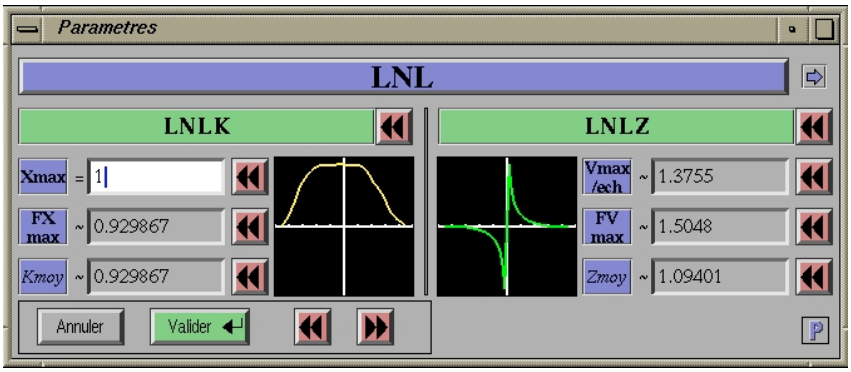


Figure 38 : prévisualisation et édition de l'amplitude des caractéristique des LNL dans la fenêtre des paramètres de GENESIS 1.5 *(couleur p.446).*

L'ergonomie est similaire à celle que nous avons exposée plus haut. Elle respecte, notamment, les principes d'homogénéité et de validation. S'y ajoute, cependant, la possibilité d'ouvrir la fenêtre d'édition profonde en cliquant sur l'une des deux caractéristiques représentées.

14.3.2 - La fenêtre d'édition profonde

Cette seconde fenêtre est modale (ou bloquante). Cela se justifie dans la mesure ou elle suppose un changement d'objet d'intérêt (on passe de l'objet CORDIS-ANIMA aux caractéristiques de l'un de ses modules) et une modification du mode de pensée.

Dans cette fenêtre, l'utilisateur a toute liberté pour spécifier la forme des caractéristiques. Pour cela, deux modalités complémentaires sont proposées :

- L'édition par manipulation directe des poignées de la courbe. Ici, nous avons porté une attention particulière à ce que les fonctionnalités de manipulation directe (insertion, suppression, déplacement, navigation…) soient *cohérentes* avec celles qui se pratiquent sur les objets dans l'établi.
- L'édition alphanumérique des positions des poignées. L'ergonomie est similaire à celle de la fenêtre des conditions initiales : l'utilisateur sélectionne d'abord les poignées qui l'intéressent, dont les coordonnées sont alors listées dans un tableau. Il peut ensuite, poignée par poignée, entrer au clavier les valeurs qui l'intéressent.

La coexistence de ces deux modalités offre à la fois la liberté et l'intuition propres à la manipulation directe et la précision permise par l'édition alphanumérique.

Enfin, précisons que c'est dans cette fenêtre que l'utilisateur a la possibilité de spécifier les propriétés de symétrie de la caractéristique ainsi que les types d'interpolation à utiliser entre chaque poignée (linéaire, spline ou hyperbolique).

La figure 40 page suivante résume, pour terminer, les flux d'interface de ces deux fenêtres.

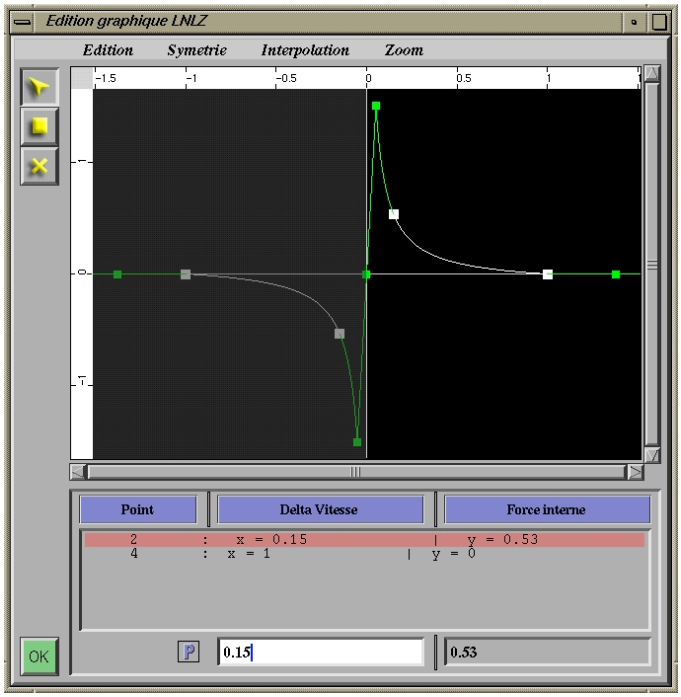


Figure 39 : La fenêtre bloquante pour l'édition des courbes des LNL *(couleur p.446).*

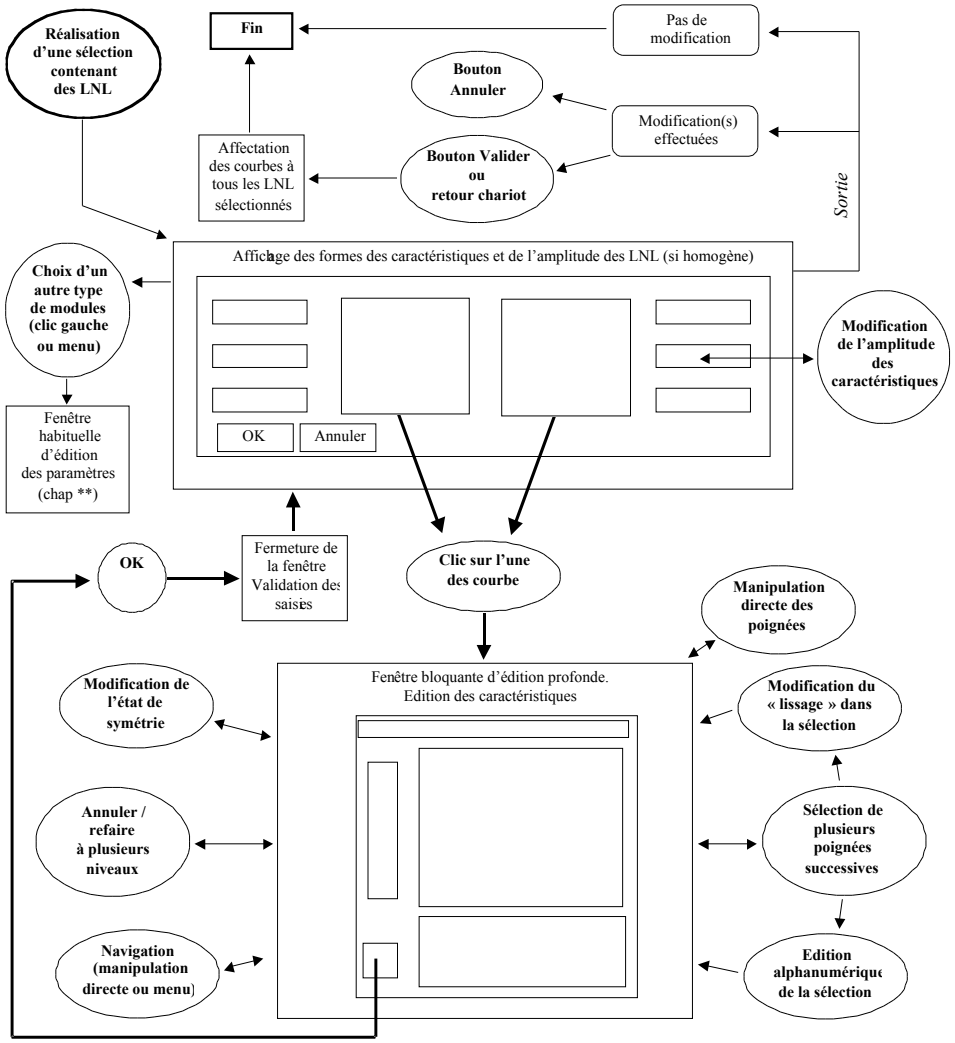


Figure 40 : flux d'interface pour l'édition des LNL

Chapitre 15

PERSPECTIVES POUR LES FONCTIONNALITES FONDAMENTALES

Les nombreux cas d'utilisation de GENESIS que nous avons pu observer et le retour des utilisateurs permettent de considérer que les fonctionnalités fondamentales de GENESIS et leur ergonomie sont satisfaisantes et stabilisées. Elles appellent cependant quelques remarques et critiques, et suggèrent de nouveaux développements. Nous en évoquons les principaux dans ce dernier chapitre[1]. Les propositions qui y figurent sont essentiellement prospectives. Certaines d'entre elles ont cependant déjà été implémentées sous la forme de prototype.

15.1 - Echelles et valeurs par défaut

Dans GENESIS, les valeurs par défaut des paramètres utilisées lorsqu'un module est ajouté sont :

M = 1 K = 0.01 Z = 0.0001 S=0 (pour les BUT)

Ces valeurs sont satisfaisantes lorsque l'utilisateur travaille sur une structure vibrante : elles assurent, en général, que l'objet construit oscille dans la plage des fréquences audio.

Par contre, dès lors que l'utilisateur s'intéresse à un objet d'un autre type dans la hiérarchie des structures, mettons un chef, l'ajout d'un module est suivi de l'affectation de ses paramètres. Deux actions élémentaires sont donc systématiquement enchaînées pour la réalisation d'une tâche unique. De plus, les utilisateurs novices ne connaissent pas les plages de valeurs qu'il convient alors d'utiliser. Leur travail est considérablement freiné, voire stoppé.

L'observation d'utilisateurs avertis en situation de création est ici intéressante.

Elle montre que la valeur des paramètres est dans un premier temps choisie de telle sorte que le comportement de l'objet s'approche de celui souhaité, sans plus de précision. Qui plus est, la valeur affectée est presque systématiquement celle de l'un des *points d'accumulation* qu'on peut observer statistiquement pour le paramètre considéré dans les modèles (voir figure 14 p240 plus haut)[2].

En se basant sur ces observations, il est possible de dégager les grandes lignes de ce que pourraient devenir, dans un proche avenir, le système des valeurs par défaut dans GENESIS.

Ce système suppose que l'utilisateur puisse d'une part spécifier pour chaque paramètre l'ensemble des *valeurs par défaut* qu'il veut pouvoir utiliser (GENESIS proposant par défaut un ensemble de base) et d'autre part choisir, parmi celles-ci, la (ou les) valeurs à utiliser lorsqu'il ajoute un module.

[1] Chacune des parties qui suivent proposera un tel chapitre consacré aux perspectives.

[2] Rappelons toutefois que si certaines de ces valeurs préférées des paramètres sont partagées par tous les utilisateurs, d'autres sont spécifiques à chaque utilisateur.

Quelques directions peuvent être données quant à l'ergonomie de ces fonctionnalités.

Tout d'abord, l'ensemble des *valeurs par défaut* devraient correspondre à l'ensemble des classes utilisées pour la représentation des paramètres sur l'établi. L'idée est donc que pour chaque classe que l'utilisateur définit pour la représentation des modules (dans la fenêtre des préférences) il ajoute une *valeur par défaut* ou *valeur préférée* correspondante.

Ensuite, il paraît possible d'associer le système aux notions d'échelle et de hiérarchie des sous-objets. En effet, il semble qu'il soit possible d'associer un *type* d'objet à chaque triplet de valeurs (M,K,Z) construit sur l'ensemble des valeurs initiales que l'utilisateur a l'habitude d'utiliser.

Par exemple, lorsqu'il veut un chef il choisit :

$M = 1e^{10}$	K = 1	Z = 0.01	S=0 (pour les BUT)

Et lorsqu'il veut un instrumentiste :

M = 1000	K = 0.01	Z = 0.01	S=0 (pour les BUT)

Aussi, nous proposons que l'utilisateur puisse explicitement associer une *échelle* et un nom à chaque triplet (classe de M, classe de K, classe de Z) et donc à chaque triplet de valeurs préférée.

Ceci-fait, il pourrait à tout instant préciser l'échelle à laquelle il veut travailler – par exemple dans un nouvel espace de la palette d'outil. Les paramètres utilisés pour l'ajout de modules seraient alors ceux de l'échelle choisie. Précisons, incidemment, que l'impact de l'échelle choisie dans l'interface devrait pouvoir à terme être étendu telles de telle sorte qu'elle affecte les valeurs nominales des zooms, les options de sélection, etc.

L'ensemble des propositions exposées dans ce paragraphe doivent être envisagées avec circonspection. Il est certain que la notion d'échelle est essentielle au processus de création avec GENESIS. C'est cependant, comme nous l'avons vu, une notion complexe qui se décline de multiples façons[1]. Jusqu'à GENESIS 1.5, nous avons par prudence décidé de ne pas l'expliciter dans l'interface, et il n'est pas certain que les fonctionnalités décrites plus haut soient en la matière optimale. C'est qu'elles doivent *soutenir* et *faciliter* les tâches de l'utilisateur sans pour autant *restreindre* d'une quelconque manière la créativité. Aussi, nous considérons qu'avant d'être définitivement introduites dans GENESIS elles doivent être réfléchies plus avant et testées attentivement.

[1] Voir la partie III, paragraphe 8.5.

15.2 - Extension de la manipulation directe et de la représentation

Nous retenons trois voies pour une future extension de la représentation des objets et les fonctionnalités de manipulation directe.

La première consisterait à ce que les actions réalisables par manipulation directe, notamment la sélection de modules, soient conditionnées au taux de zoom.

Nous avons expliqué comment la hiérarchie des sous objets est progressivement révélée au fur et à mesure que le zoom diminue et inversement comment l'utilisateur effectue un zoom avant lorsqu'il veut travailler des objets de plus en plus bas dans la hiérarchie. L'interface, à ce jour, ne tire pas pleinement bénéfice de ces propriétés. Une idée simple et qui a déjà donné lieu à quelques expérimentations prometteuses serait qu'un clic de la souris ne sélectionne plus les modules isolément, mais d'un seul coups tous les modules qui, compte tenu du taux de zoom, apparaissent *superposés*. Pour un zoom arrière important les modules qui se trouvent dans la même zone de l'établi et qui donc, en général, appartiennent au même sous-objet, voire à plusieurs sous objets, ne pourraient être sélectionnés qu'ensemble. Pour isoler un objet, puis un module seul parmi ceux-ci, l'utilisateur devrait alors effectuer un zoom avant jusqu'à ce l'objet ou le module ne soit plus *superposé* à ses voisins.

Ainsi, à un taux de zoom correspondrait non seulement un niveau de visualisation de la hiérarchie des sous-objets mais encore un niveau de modularité accessible à la manipulation directe : modules, objet simple ou groupe d'objets.

La deuxième voie que nous retenons a trait à l'organisation ou plutôt la réorganisation des modules sur l'établi. Seules les fonctionnalités essentielles de manipulation directe existent à l'heure actuelle et il serait intéressant pour l'utilisateur de disposer de quelques outils plus évolués. Ici, il s'agit essentiellement de définir des fonctionnalités permettant une réorganisation automatique des modules MAT, en s'inspirant entre autres des utilisations typiques de l'espace de représentation qu'on peut constater. A minima, il conviendrait de concevoir un outil de réorganisation capable d'étudier le topologie du réseau pour en proposer une représentation « optimale », c'est-à-dire qui affiche le réseau de la façon la plus claire et la plus régulière possible. En complément, il serait utile que les modules MAT puissent être automatiquement replacés sur les nœuds de la grille qui quadrille l'établi.

La troisième voie que nous évoquerons s'inscrit dans la continuité de la précédente. Elle vise cependant plus spécifiquement une représentation des paramètres.

Nous avons vu pourquoi il est nécessaire, face à la version topologique de CORDIS-ANIMA, d'introduire un espace de représentation plan perpendiculaire à l'axe de simulation. Nous avons par ailleurs montré que la façon dont l'objet est organisé dans plan de l'établi n'a aucun effet sur les simulations et expliqué pourquoi cette propriété a tendance à déranger les utilisateurs novices. Nous avons évoqué, enfin, les limitations des choix retenus pour la représentation des paramètres et expliqué que l'utilisateur est souvent conduit à utiliser le placement des modules dans l'établi pour faire ressortir certaines propriétés paramétriques – telle l'inhomogénéité de certains paramètres.

Face à ces différentes remarques, une voie de recherche très intéressante est d'étudier la façon dont le placement relatif des modules sur l'établi pourrait être lié à la valeur de leurs paramètres– et réciproquement.

Plusieurs problèmes se posent cependant.

Notamment, il ne peut être question de proposer ce type de fonctionnalités autrement qu'en option, puisqu'il est aujourd'hui avéré que l'établi offre une latitude très intéressante à l'utilisateur en lui permettant de faire ressortir toutes sortes d'informations symboliques.

Par ailleurs, il est bien certain que les avantages, limites et risques d'une telle fonctionnalités doivent être pesés avec une précaution particulière tant elle peut influer sur l'utilisabilité de GENESIS et plus encore sur le modèle mental induit par son utilisation.

Face à ces enjeux et problèmes, une première piste originale et intéressante consisterait à définir, dans l'établi, un espace physique CORDIS-ANIMA 2D.

Deux types de simulations seraient alors possibles dans GENESIS. Le premier, traditionnel, consisterait à utiliser l'axe de simulation orthogonal. Le second consisterait alors à simuler l'objet dans l'espace 2D, avec des paramètres particuliers déduits des paramètres effectifs des modules topologiques.

Le nouveau placement des modules qui serait obtenu à l'issue de la simulation serait alors, d'une marnière ou d'une autre, dépendant de leurs paramètres. On peut espérer qu'il fasse ressortir certaines de leurs propriétés – par exemple les élasticités importantes qui tendraient à rapprocher dans l'établi les <MAT> connectés, les inerties faibles, etc. Ici, l'essentiel du travail consiste précisément à définir la façon dont les paramètres du modèle 2D doivent être calculés à partir des paramètres effectifs des modules. C'est ce lien de dépendance, en effet, qui conditionnera la pertinence du replacement obtenu et l'intérêt global de cet outil de réorganisation original.

On comprendra, cependant, qu'il y a ici matière à des investigations conséquentes, et ce d'autant plus que dans un deuxième temps on peut imaginer que les paramètres des modules soient à leur tour affectés lorsque l'utilisateur modifie l'organisation des MAT sur l'établi. La réflexion concernant ce type de fonctionnalités a à peine débuté et l'idée, pour être séduisante, doit être confrontée à l'expérience.

15.3 - Trans-représentation et multimodalité

La représentation graphique et les fonctionnalités de manipulation directe des objets proposées par GENESIS 1.5 soutiennent la simplicité et le naturel du modèle mental possible face formalisme CORDIS-ANIMA. Elles tendent à ce que l'utilisateur envisage ses objets comme des objets réels – représentés, certes – plutôt que comme des modèles plus abstraits et l'aide à développer une certaine intuition face au processus de lutherie. Elles atteignent, en conséquence, les principaux enjeux que nous leur avions attribués.

Cependant, au fur et à mesure que les exigences entretenues vis à vis de GENESIS augmentaient, il est apparu qu'un style d'interaction permettant d'opérer dans un domaine plus symbolique ou abstrait pourrait apporter dans certains cas une aide précieuse.

Ce n'est d'ailleurs guère étonnant : dans la partie II, nous avons rappelé que la manipulation directe ne permettait guère certaines opérations, et qu'inversement l'interaction via un langage a ses avantages. Dans GENESIS, le fait de disposer d'un langage pour la conception des objets permettrait par exemple de réaliser aisément

certaines actions répétitives telle l'assemblage d'une structure à l'aide de boucles. Il autoriserait également l'insertion par référence (duplication d'un objet avec conservation d'un lien de descendance avec l'objet « père »). De façon plus générale, il favoriserait une approche plus symbolique du processus de lutherie ou de modélisation. Enfin, nous verrons dans la suite que certaines fonctionnalités d'ores et déjà prévues – et parfois prototypées – tel un système de relation et méta-paramètres, prélude aux fonctionnalités d'édition évoluées des paramètres – nécessitent qu'un langage soit défini, et que la manipulation directe soit au moins momentanément abandonnée.

Une première étape simple et d'ambition très limitée a déjà été franchie en direction d'une trans-représentation des objets avec la possibilité d'exporter et d'importer des fichiers dans un format textuel. Ces fichiers .lst ont été conçus pour être lisibles au sein d'un tableur. Ainsi, il est possible de disposer d'un ensemble de fonctionnalités très différentes de celles que propose GENESIS – encore que ce soit hors de l'environnement. L'avenir devrait montrer si une telle *mise à plat* du contenu des objets présente un intérêt fort. Dans l'affirmative, il serait envisageable de proposer au sein même de GENESIS des fonctionnalités similaires.

On obtiendrait alors deux modes pour l'édition des paramètres et des conditions initiales. Le premier reposerait sur les fenêtres présentées plus haut et donc notamment sur le principe de l'homogénéité pour les paramètres physiques et sur une interactivité maximale avec l'établi. Le second, basé sur une ergonomie de type tableur, permettrait lorsque c'est intéressant de quitter la représentation habituelle des objets pour développer un mode de pensée légèrement différent.

Ce type de développement est envisageable à court terme. Il ne constitue, par contre, qu'une toute première approche face au problème qui nous préoccupe.

Compte tenu des avantages cruciaux de la manipulation directe et de la représentation graphique des objets dans le contexte de GENESIS, nous considérons qu'elles doivent demeurer au cœur de l'interface. Ainsi, l'objectif à terme serait que deux styles d'interaction très différents (*direct* et *par un langage*) cohabitent dans GENESIS. On adresse ici la question intéressante et difficile des interfaces multimodales, actuellement au cœur de plusieurs recherches relevant de l'Interface Homme-Machine.

Partant de la manipulation directe, un élément de solution consisterait à enregistrer à l'aide d'un langage toutes les modifications appliquées à l'objet par l'utilisateur (ajouter un module, modifier un paramètre, etc.), de telle sorte que les textes ainsi générés rendent compte de sa genèse. Mais ce n'est là qu'une première étape. Il faudrait ensuite que cet « historique » puisse être analysé pour que de la série d'actions enregistrées soit inférée une cohérence de plus haut niveau – des boucles, par exemple, venant remplacer les actions répétées de l'utilisateur.

Réciproquement, il deviendrait nécessaire que des objets travaillés à l'aide du langage, incluant par exemple de telles boucles de construction, deviennent ensuite accessibles à la manipulation directe, sans pour autant que la cohérence initiale ne soit détruite.

Le principe d'une interface multimodale pour GENESIS est donc certes séduisant, mais il apparaît que sa mise en œuvre est difficile. De façon plus générale, si ils ont conduit à la définition d'un ensemble de fonctionnalités fondamentales globalement satisfaisant, ils ne constituent pourtant que le prélude à de plus amples recherches. Le travail, donc, devra se poursuivre.

RESUME ET CONCLUSION

Cette partie a donc introduit les principes et fonctionnalités de base de l'interface GENESIS :

- La notion d'établi et d'espace de représentation et les choix quant à la représentation des modules, des objets et des paramètres ;
- La manipulation directe pour l'édition des réseaux MAT/LIA qui constituent les objets, ainsi que les fonctions de navigation dans l'établi ;
- Les fenêtres d'édition des paramètres et des conditions initiales.

L'ergonomie de ces fonctionnalités fondamentales a été pensée pour soutenir l'utilisateur novice qui, au cours de ses premières confrontations au logiciel, doit à la fois intégrer les principes de base du formalisme CORDIS-ANIMA et se construire un *modèle mental* pertinent relativement à l'interface. Elles ont été conçues et optimisées pour satisfaire également l'utilisateur expert qui lui demande à l'interface à la fois puissance, rapidité, fiabilité et souplesse.

Avec ces seules fonctionnalités fondamentales, il est *théoriquement* possible de générer tous les objets répondant au formalisme CORDIS-ANIMA topologique, et ce dans des conditions de confort et d'efficacité optimales.

Pour autant, elles devront à l'avenir être étendues. Le quatrième chapitre de cette partie a précisément exposé les principales évolutions qui nous semblent intéressantes pour ces fonctionnalités à court ou moyen terme.

Ces fonctionnalités fondamentales sont par ailleurs insuffisantes pour faire de GENESIS un environnement complet, puisqu'elles n'apportent un soutien qu'à une faible partie des tâches que nous avons identifiées pour la phase de lutherie. Dans la partie suivante, nous abordons les fonctionnalités *évoluées* qui ont été conçues en ce sens.

Partie VI

Fonctionnalités évoluées du mode Lutherie

Cette partie est consacrée aux fonctionnalités qui, sans être dictées par les nécessités du formalisme CORDIS-ANIMA, ont été conçues pour permettre un travail de lutherie évolué. Elle présente successivement :

- Les fonctionnalités conçues pour soutenir la démarche macro-modulaire de l'utilisateur ;
- L'outil d'analyse et d'accordage ;
- Plusieurs outils complémentaires, plus simples mais très utiles, qui complètent l'environnement.

Enfin, dans un quatrième chapitre, nous synthétisons les principales perspectives d'évolutions pour l'atelier de lutherie GENESIS. Ces fonctionnalités n'existent toutefois à ce jour qu'à l'état de prototype ou de projet. En les évoquant, nous complétons la présentation de ce que devrait être selon nous un mode lutherie optimal pour GENESIS.

Introduction

Contexte et bibliographie

Partie I : le paradigme de la simulation interactive et multisensorielle d'objets physiques

Partie II : Interaction Homme-Machine

Partie III : Le processus de création avec GENESIS. Analyse des Tâches

Conception de GENESIS

Partie IV : Choix généraux pour l'environnement GENESIS

Partie V : Fonctionnalités fondamentales de l'atelier de lutherie GENESIS

Partie VI : fonctionnalités évoluées du mode lutherie

Chapitre 16 Macro-modularité : Ensembles et capsules 278

16.1 - UNE ANALYSE DU PROBLÈME 278
16.2 - ENSEMBLES 281
16.3 - CAPSULES 284
16.4 - CONCLUSION 287

Chapitre 17 Analyse modale et accordage 288

17.1 - APPROCHE THÉORIQUE DE L'ANALYSE MODALE 288
17.2 - L'ANALYSE MODALE DANS GENESIS 298

Chapitre 18 Outils complémentaires d'édition 303

18.1 - GÉNÉRATION DE STRUCTURES RÉGULIÈRES 303
18.2 - SUBSTITUTION DE MODULES 305
18.3 - MULTIPLICATION DES PARAMÈTRES ET MAÎTRISE DES ÉCHELLES 306

Chapitre 19 Perspective pour les fonctionnalites evoluees 308

19.1 - EDITION ÉVOLUÉE DES PARAMÈTRES ; RELATIONS ET META-PARAMÈTRES 308
19.2 - EVOLUTION DES FONCTIONNALITÉS DE MACRO-MODULARITÉ 313
19.3 - AUTRES FONCTIONNALITÉS 314

Résume et Conclusion 316

Partie VII : Simulation et phénomènes

Conclusions et perspectives

Chapitre 16

MACRO-MODULARITE : ENSEMBLES ET CAPSULES

Le 'niveau des modules' constitue la base de GENESIS. Il permet d'expérimenter et de maîtriser les briques indivisibles définies par le formalisme, qui toutes modélisent un comportement élémentaire de la matière. Il est rare, cependant, qu'un module puisse être envisagé isolément. C'est au contraire en travaillant des inter-relations entre *plusieurs* modules que l'utilisateur confère du sens aux objets qu'il conçoit.

Nous appelons *démarche macro-modulaire* l'attitude qui consiste précisément à s'abstraire du niveau des modules élémentaires pour considérer des ensembles regroupant plusieurs d'entre eux. La démarche macro-modulaire tient une place essentielle dans la phase de lutherie. Tout travail conséquent avec CORDIS-ANIMA y fait nécessairement appel.

Jusqu'à la version 1.4 de GENESIS, la construction macro-modulaire des objets restait tacite puisqu'il n'était pas possible d'introduire dans les objets des informations qui la représenteraient en tant que telle. Il est pourtant très utile pour l'utilisateur de pouvoir expliciter l'organisation macro-modulaire qui guide son travail. Ce chapitre est consacré aux fonctionnalités qui ont été conçues en ce sens.

- Dans un premier temps, nous rappelons les enjeux multiples de la macro-modularité dans GENESIS ;
- Dans un second temps, nous introduisons les notions *d'Ensemble* et de *Capsule* que nous avons défini pour répondre à ces enjeux, ainsi que leur implémentation dans GENESIS 1.5.

16.1 - Une analyse du problème

16.1.1 - Enjeux

La notion de macro-modularité se rencontre dans toutes les disciplines élaborant des fonctions complexes à partir de briques de base. De façon générale, il s'agit de substituer à un ensemble complexe de 'briques' une entité unique qui les réunissent toutes.

Dans GENESIS, la sélection offre un premier soutien à la démarche macro-modulaire. Il serait cependant fort utile qu'un ensemble de modules puisse être explicitement identifié comme un macro-module ou 'sous-objet'. Ainsi, les enjeux traditionnels de la macro-modularité s'appliquent-ils au cas de GENESIS. Il s'agit de permettre :

- Une réduction de la charge cognitive, d'abord, étant entendu qu'il est plus aisé de manipuler une entité cohérente qu'un ensemble complexe d'entités plus simples[1] ;
- A terme, un changement d'échelle dans le processus de création, puisque alors des objets complets (cordes, membranes, plaques…) pourraient être manipulés à la manière de modules élémentaires.

[1] On rejoint ici la question de la mémoire à court terme ou mémoire de travail : sa capacité est limitée, mais chaque unité d'information ou 'chunk' peut être complexe, l'essentiel étant qu'elle constitue un tout cohérent pour le sujet – voir la partie IV.

Mais, entre le fait de sélectionner quelques modules pour affecter des valeurs à leurs paramètres et la notion de sous-objet, il existe tout un jeu de nuances possibles. La conception des fonctionnalités macro-modulaires adresse une importante diversité de situations et d'enjeux qu'il nous faut bien comprendre. Nous distinguons quatre enjeux spécifiques à l'environnement GENESIS.

Faciliter la sélection et la navigation

L'élection d'une sélection est une tâche particulièrement récurrente dans GENESIS. Nous avons discuté plus haut les options qui permettent un travail aisé de la sélection par 'manipulation directe'. Ces options, cependant, ne traitent pas la situation courante où plusieurs modules doivent être sélectionnés de façon répétitive. Une telle situation demande que la liste des modules sélectionnés puisse être consignée pour une re-sélection ultérieure. Il s'agit ici en outre de faciliter la navigation dans les objets complexes en aidant la recherche des différentes parties qui les constituent.

Nommer. Organiser. Communiquer.

Les utilisateurs de GENESIS ont tous expérimenté la difficulté d'analyser un objet pré-conçu – par exemple extrait d'un catalogue – pour le réutiliser[1]. Cette difficulté vaut d'ailleurs bien au delà de GENESIS [2]. De façon générale, la lecture des schémas suppose une stratégie qui s'appuie notamment sur la reconnaissance de 'blocs' fonctionnels que l'on décompose ensuite. L'un des enjeux des fonctionnalités macro-modulaires est de guider la découverte des objets. Cela suppose, à minima, que certains ensemble de modules puissent être nommés et commentés.

Associer un traitement à un ensemble de modules

De nombreux traitements possibles dans GENESIS concernent plusieurs modules. C'est par exemple le cas de *l'analyse modale* et plus généralement de toute analyse des propriétés d'un objet ou encore des simulations lorsqu'elles ne concernent qu'une partie seulement d'un objet. Les fonctionnalités macro-modulaires doivent permettre de conserver la liste des modules auxquels un tel traitement a été appliqué.

Protéger ; réduire la variabilité

Lorsqu'un certain degré de finalisation est atteint, il devient important de pouvoir protéger certaines dimensions de l'objet. La conception des fonctionnalités de protection demande une analyse fine des motivations sous-jacentes. Plusieurs niveaux de protection sont envisageables : protection structurelles contre la suppression accidentelle ou la connexion / déconnexion des modules LIA, protection des paramètres, protection des relations de proportionnalités entre paramètres, etc.

La protection ne concerne ni le niveau des modules (il n'est que rarement utile de protéger un module isolé) ni l'objet complet, mais plutôt certains ensembles de modules. C'est donc au niveau des fonctionnalités macro-modulaires qu'elle doit être envisagée.

La protection n'a pas pour seul intérêt d'éviter les erreurs de manipulation. Elle permet aussi de réduire la *variabilité* des objets et de façon corollaire de guider les variations possibles, tant paramétriques que structurelles. L'enjeu, ici, est qu'avec les

[1] Plusieurs utilisateurs se sont exprimés sur ce point – voir l'annexe D.

[2] Il est toujours plus facile d'élaborer un schéma (un 'pattern iconique') que de le lire et le comprendre Ce point est discuté dans le chapitre 3. Voir également [Kay.90].

fonctionnalités macro-modulaires l'utilisateur puisse flécher les variations intéressantes, indiquer clairement les dimensions qu'il juge pertinent de faire varier lorsqu'il s'agit de décliner un objet en une famille instrumentale.

16.1.2 - Typologie des ensembles de modules dans GENESIS

A la lumière de ce qui précède et des observations d'utilisateurs in-situ, nous proposons dans ce paragraphe une typologie simple des ensembles de modules qui peuvent apparaître dans les objets CORDIS-ANIMA. Nous retenons, pour ce faire, deux caractères : le premier est relatif aux propriétés internes des ensembles, le second à leur permanence.

Macro-modularité fonctionnelle et morphologique

Nous distinguerons, d'abord, la macro-modularité fonctionnelle et la macro-modularité morphologique.

Un objet CORDIS-ANIMA est un objet hiérarchisé, constitué de plusieurs sous-objets dont chacun a une fonction. On dira qu'un ensemble *fonctionnel* ou *sous-objet* est constitué lorsque il a une *fonction* identifiable dans l'objet global et qu'isolé il garde une cohérence physique. Un ensemble fonctionnel peut être, par exemple, une metaphore de corde, un instrument constitué de plusieurs cordes et de chevalets, un instrumentiste, etc. La macro-modularité fonctionnelle est une première catégorie essentielle au sein des objets CORDIS-ANIMA. Ce n'est pas la seule.

Il est souvent utile, en effet, d'isoler plusieurs modules sans pour autant que l'ensemble ainsi constitué n'ait une quelconque signification physique. On dira alors qu'un ensemble *morphologique* est créé. Un ensemble *morphologique* pourrait-être, par exemple, les modules MAS homogènes d'un sous-objet, l'ensemble des chevalets de l'objet complet qu'il peut être intéressant d'éditer conjointement ou encore un déclencheur et le module connecté du sous objet déclenché.

Permanance ; macro-modularité faible et forte

Le second caractère que nous retenons tient à la *permanence* des ensembles. Nous distinguerons alors la macro-modularité forte et la macro-modularité faible.

Nous dirons qu'un ensemble de modules est *faible* lorsqu'il n'est que momentanément nécessaire à la réalisation d'une tâche particulière mais que l'utilisateur souhaite conserver par ailleurs un accès individuel aux modules qui le constituent. Par exemple, un ensemble faible existe lorsque l'utilisateur veut modifier conjointement les paramètres de plusieurs modules homogènes mais conserver la possibilité d'en modifier le réseau.

Nous dirons que l'utilisateur a créé un ensemble *fort* ou *macro-module* dès lors qu'il souhaite envisager *globalement* certains modules sans plus avoir accès à chacun d'entre eux. Notons qu'un tel macro-module doit, par nature, faire apparaître plusieurs informations à sa *surface*. On peut d'ores et déjà distinguer la surface *structurelle* constituée de l'ensemble de ses points de connexion et la surface *paramétrique* qui regroupe l'ensemble des paramètres accessibles. Le fait d'identifier un ensemble *fort* limite la *variabilité* à la fois en termes structurels (la structure du réseau MAT/LIA de l'ensemble est figée) et en termes de paramètres (les variations paramétriques possibles dépendent de la surface paramétrique retenue). Du même coup cependant, les ensembles forts permettent de diminuer la difficulté de compréhension et de manipulation – disons, la charge cognitive – et c'est là probablement l'un de leurs intérêts majeurs.

16.1.3 - Deux notions, deux fonctionnalités

La diversité des enjeux de la macro-modularité et des types d'ensembles qu'il est intéressant d'expliciter au sein des objets rend difficile la définition d'une fonctionnalité unique. Plus, elle tend à ce que plusieurs notions et fonctionnalités soient proposées, l'utilisateur étant libre de choisir parmi elles en fonction de ses objectifs. GENESIS 1.5 définit les deux notions complémentaires d'*Ensemble* et de *Capsule*. La suite de ce chapitre les présente tour à tour. Dans un dernier paragraphe, nous montrons comment Ensembles et Capsules se complètent pour répondre à la diversité des situations possibles lorsqu'il s'agit d'expliciter, dans un objet, les relations structurelles entre plusieurs modules.

16.2 - Ensembles

16.2.1 - Définition et utilisations

Un Ensemble est définit par son *étiquette* et par son *contenu*. L'étiquette comprend toujours un nom et un commentaire et, le cas échéant, des informations quant au niveau de protection[1]. L'étiquette peut également renvoyer à d'autres types d'informations, tels les résultats d'un traitement effectué sur le contenu de l'Ensemble. Le contenu de l'Ensemble est une simple liste de modules CORDIS-ANIMA figurant sur l'établi.

Etiquette : nom + commentaire + informations complémentaires
{Nom} : contenu, liste de modules

Figure 1 : définition d'un ensemble

Les Ensembles sont non-exclusifs : un module peut appartenir à plusieurs Ensembles, chacun d'entre eux permettant de l'envisager dans un contexte spécifique. Par exemple, un déclencheur peut appartenir à la fois à l'Ensemble des déclencheurs de l'objet et à l'Ensemble constitué de la structure vibrante déclenchée et des éventuels instrumentistes qui lui sont connectés. Le premier Ensemble est alors utile pour éditer les conditions initiales des déclencheurs de l'objet (i.e. : la 'partition de déclencheurs') et le second pour déplacer d'un seul coup la structure et ses déclencheurs sur l'établi. Les Ensembles constituent donc un moyen efficace pour expliciter l'existence d'une relation structurelle entre plusieurs modules, sans présumer de la signification de cette relation.

16.2.2 - Identification des tâches relatives aux Ensembles

Deux catégories bien distinctes de tâches coexistent autour de la notion d'Ensemble telle que nous l'avons définie : d'une part celles qui relèvent de leur *définition* d'autre part celles qui relèvent de leur *utilisation.*

Définir un ensemble, c'est essentiellement attribuer une étiquette à plusieurs modules sélectionnés. Utiliser un Ensemble, à l'inverse, c'est sélectionner les modules qu'il contient par le truchement de son étiquette. A ces tâches essentielles s'ajoutent plusieurs tâches dérivées résumées dans le tableau de la figure 2.

[1] La protection des Ensembles n'est pas implémentée dans GENESIS 1.5. Elle est discutée dans le chapitre 4 de cette partie.

Définition des Ensembles	Utilisation des Ensembles.
Créer un nouvel Ensemble	*S'informer* Savoir quand {Ensemble} = {Sélection} Savoir quand {Ensemble} ⊄ {Sélection}
Supprimer un Ensemble	
Modifier l'étiquette d'un Ensemble	
Modifier le contenu d'un Ensemble {Ensemble} = {Sélection} {Ensemble} = {Ensemble} U {Sélection} {Ensemble} = {Ensemble} \ {Sélection}	***Travailler la sélection*** {Sélection} = {Ensemble} {Sélection} = {Sélection} U {Ensemble} {Sélection} = {Sélection} \ {Ensemble}
Organiser les étiquettes	*Copier le contenu d'un Ensemble*

Figure 2 : identification des tâches relatives aux Ensembles[1]

16.2.3 - Ergonomie

La conception d'une ergonomie optimale pour la gestion des Ensembles est rendue malaisée par l'existence de ces deux catégories de tâches. En effet un Ensemble peut être désigné de deux façons suivant qu'il s'agit de sélectionner son contenu sur l'établi ou de sélectionner son étiquette (i.e. : l'Ensemble en tant qu'objet) pour lui appliquer certaines actions.

L'ergonomie de la fenêtre des Ensembles dans GENESIS 1.5

Nous avons adoptée une solution globale qui regroupe dans une seule fenêtre *libre* et de *petite taille* (pour qu'elle puisse rester affichée sans nuire à l'organisation de l'écran) toutes les actions possibles sur les Ensembles[2]. La fenêtre de gestion des Ensembles proposée par GENESIS 1.5 est présentée sur la figure 3. Elle fait apparaître trois zones distinctes.

- Le menu. Il regroupe toutes les fonctionnalités relatives à la définition des Ensembles.
- La colonne de gauche. Elle est dédiée au travail de la sélection. Les boutons circulaires représentent le *contenu* des Ensembles. Un clic sur un tel bouton sélectionne le contenu de l'ensemble concerné sur l'établi. Ces boutons servent en outre au retour d'information. Leur couleur reflète, à tout instant, le statut des différents Ensembles relativement à la sélection : lorsque la sélection est égale au contenu d'un Ensemble, le bouton correspondant est vert. Lorsque le contenu est simplement inclus dans la sélection, le bouton est bleu. Dans tous les autres cas, il apparaît grisé.

[1] On pourrait ajouter à ces tâches des opérations inter-Ensembles (fusionnement, etc). Il s'avère cependant plus avisé, compte tenu de l'organisation générale de GENESIS, d'imposer un recours systématique à la sélection. Notons par ailleurs que dans ce tableau ne figure aucune information quant à l'édition des niveaux de protection, ou à celle des données afférentes. La protection des Ensembles n'est pas à ce jour implémentée et sera discutée dans le chapitre 4. Quant aux données afférentes, elles ne sont pas, dans GENESIS, directement accessibles via les Ensembles, mais dans les fenêtres qui leur sont propres, par le truchement de la sélection.

[2] Un prototype plus simple proposait deux fenêtres, l'une pour l'utilisation des ensembles et l'autre, modale, pour leur définition. Une telle ergonomie s'est avérée peu satisfaisante en ce qu'elle introduit plusieurs 'objets d'interface' autour de la même notion et qu'elle induit une séparation nette entre utilisation et définition, alors même que ces deux catégories de tâches sont imbriquées dans le processus de l'utilisateur.

- La colonne de droite. Elle est dédiée à la manipulation des étiquettes. Plusieurs étiquettes peuvent être sélectionnées simultanément pour qu'un traitement commun leur soit appliqué (supprimer les Ensembles, modifier l'ordre des Ensembles dans la liste par manipulation directe, modifier le contenu des Ensembles, etc).

La fenêtre des Ensembles fait donc bien apparaître deux notions de sélection : celle du contenu (sélection indirecte des modules de l'établi) et celle de l'étiquette (manipulation des Ensembles en tant qu'objet). Notons, enfin, qu'un menu contextuel est proposé à l'utilisateur sous le bouton droit de la souris. Lorsque le bouton de contenu est cliqué, ce menu permet de modifier le contenu de l'Ensemble désigné. Lorsque l'étiquette est cliquée, ce sont les manipulations des étiquettes qui sont proposées.

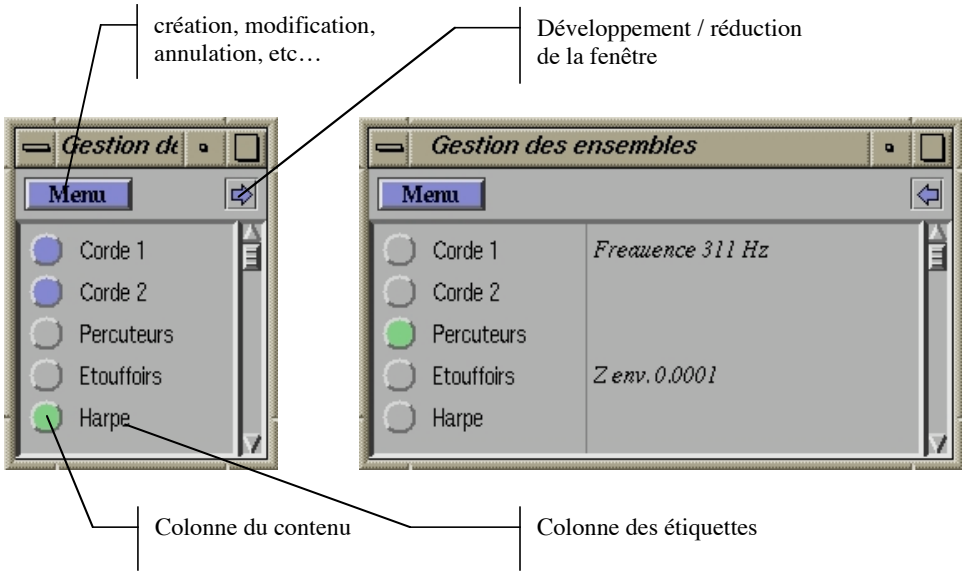


Figure 3 : Les deux aspects de la fenêtre des Ensembles de GENESIS 1.5
(couleur p.447)

Gestion des Ensembles sur l'établi

Dans GENESIS 1.5, les Ensembles sont accessibles depuis l'établi par le *double clic*, ce qui incidemment est conforme aux usages des logiciels courants puisqu'il s'agit *d'étendre* une sélection. Un double clic sur un module sélectionne donc le contenu de l'Ensemble auquel il appartient. Dans le cas ou le module appartient à plusieurs Ensembles, un menu permet de choisir celui qui était souhaité.

Analyse critique des choix opérés

N'ayant pas encore été testés in situ, les Ensembles sont appelés à être affinés à l'avenir en bénéficiant du retour d'utilisateurs. Nous estimons cependant que les choix ergonomiques opérés condensent dans un espace de taille réduite toutes les fonctionnalités concernées et assurent en conséquence que le potentiel des Ensembles soit tout entier 'à portée de la main' sans nuire à la lisibilité ou l'organisation de l'écran. A ce titre ces choix devraient donner satisfaction.

Plusieurs compléments devront être apportés à cette première approche. Il est notamment prévu d'ajouter à la version actuelle les possibilités de *protection* évoquées plus haut. Ceci sera discuté au chapitre 19.

16.3 - Capsules

Les Ensembles permettent avant tout de *nommer* certaines parties de l'objet et de *travailler la sélection*. Ils ne répondent pas à tous les enjeux de la macro-modularité. Même protégés, ils n'offriraient pas toute la puissance de la notion de macro-module. Les *capsules* ont précisément été pensées pour les compléter en ce sens.

16.3.1 - Définition

Nous proposons tout d'abord une définition complète des capsules qui prend en compte toutes les fonctionnalités qui devront à terme figurer dans GENESIS. La mise en œuvre effectivement réalisée dans la version 1.5, que nous présentons ensuite, ne recouvre cependant pas encore totalement cette définition.

Les Capsules sont des objets

Une Capsule est, pour l'utilisateur et au sein du code source de GENESIS, un objet complet. Une capsule peut donc être enregistrée dans un fichier .top. Elle est dotée d'un nom, de plusieurs modules qui peuvent être visualisés dans un établi, et comprend le cas échéant plusieurs Ensembles[1] ou encore d'autres capsules.

Lorsqu'elle est insérée dans un objet, une capsule se comporte comme un module élémentaire : elle peut être sélectionnée comme un module, recevoir des connexions, être déplacée sur l'établi, paramétrée dans la fenêtre de paramètres, etc. La capsule, cependant, n'est alors visible et accessible que par sa *surface.* Deux types d'informations structurent la surface d'une capsule : des informations de nature *structurelle* et des informations de nature *paramétrique.*

Surface structurelle

La surface structurelle d'une capsule comprend l'ensemble des points M pouvant recevoir des connexions et celui des points L en instances de connexion (figure 4)[2].

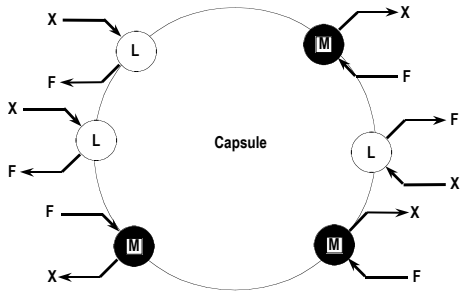


Figure 4 : la surface structurelle d'une capsule est l'ensemble des points M et des points L de la capsule susceptibles d'être connectés.

[1] Notons ici que les modules d'entrée/sortie de type SOX et SOF (et, dans les versions ad hoc de GENESIS dédiées à l'expérimentation du jeu ENX et ENF) ne peuvent pas figurer dans une capsule. Ce choix tend à ce que la capsule soit un objet physique passif, et que les modules d'interaction restent spécifiés au niveau de l'objet dans lequel la capsule est insérée.

[2] Incidemment, la notion de surface structurelle vaut aussi pour le module intégré CEL : un CEL est équivalent à une structure SOL/REF/MAS dont la surface structurelle serait limitée au seul point M du MAS.

Surface paramétrique

La notion de surface paramétrique d'une capsule n'est pas encore mise en œuvre dans la version 1.5 de GENESIS. Elle est toutefois d'ores et déjà définie.

La surface paramétrique d'une capsule regroupe l'ensemble des champs de saisie qui sont offerts à l'utilisateur pour contrôler les paramètres des modules qu'elle contient. Plusieurs possibilités devraient pouvoir être offertes :

- Laisser tous les paramètres accessibles à la surface. Les paramètres de la capsule coïncident alors avec l'ensemble des paramètres des modules qui la constituent.
- Définir des paramètres globaux simples, par exemple pour chacun des ensembles de modules homogènes dans la capsule.
- Définir des paramètres globaux complexes qui lient de manière spécifique les différents paramètres des modules contenus dans la capsule[1].

Représentation

Deux représentations des capsules sont envisageables :

- Une représentation étendue, dans laquelle la capsule apparaît comme un sous-objet complet c'est à dire où figurent l'ensemble des modules qui la constituent. Les modules de la surface de la capsule sont alors repérés par une couleur plus intense.
- Une représentation compacte ou iconique, dans laquelle la capsule apparaît comme un module simple. Le contenu de la capsule est caché ; seuls sont visibles les points M et L de sa surface.

16.3.2 - Implémentation dans GENESIS 1.5

La création d'une capsule passe par *l'encapsulation* des modules sélectionnés d'un objet. Ces modules sont supprimés de l'objet et la capsule vient les remplacer. Incidemment, toute capsule peut être *décapsulée*. Son contenu est alors inséré dans l'objet où elle figurait et la capsule est supprimée.

La représentation des capsules peut être obtenue soit sous la forme étendue soit sous la forme compacte (figure 5). Le passage de l'une à l'autre se fait via un menu contextuel accessible par clic droit sur la capsule. Les points L de la surface structurelle (extrémités des traits sortant de la capsule lorsqu'elle est en représentation compacte) ainsi que ses points M (disques figurants sur le pourtour de la capsule en représentation compacte) peuvent être connectés comme tous les autres points M et L des modules de l'objet.

[1] Cela suppose, de façon plus générale, qu'il soit possible d'exprimer dans GENESIS des relations entre paramètres. Le système des *relations* et *meta-paramètres*, encore en cours d'étude, répondra à cette nécessité. Nous l'introduirons au chapitre 19 dans cette partie.

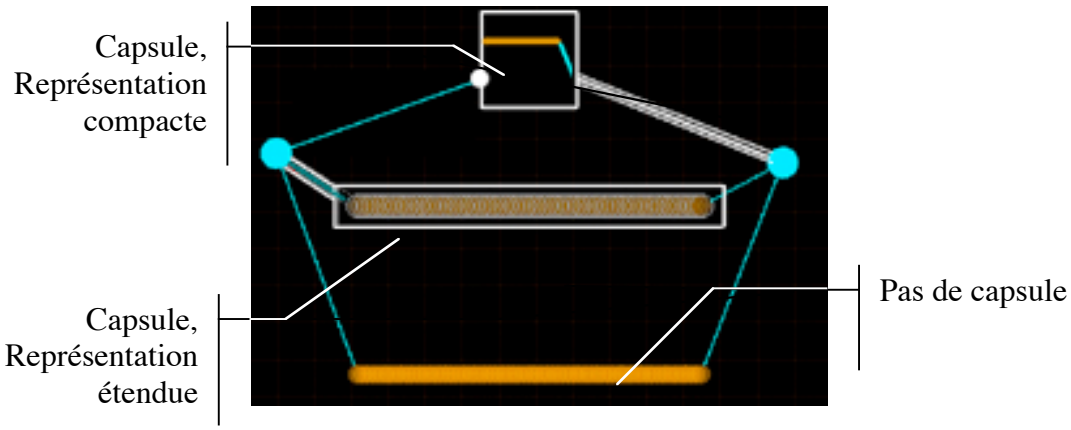


Figure 5 : capsules contenant une corde CORDIS-ANIMA en représentation compacte et étendue
(Couleur p. 447).

Enfin, toute capsule est *éditable* par double-clic. Un nouvel établi est alors ouvert, dans lequel l'objet contenu dans la capsule est visualisé. Toute capsule comprend un Ensemble SURFACE dont l'étiquette n'est pas modifiable et dont le contenu regroupe les modules de la surface de la capsule. Tout les modules non-physiques BLC en instance de connexion font partie de l'Ensemble SURFACE : ils sont appelés à être connectés aux modules de l'objet dans lequel figure la capsule. L'utilisateur peut ajouter des modules MAT de la SURFACE ou en supprimer. Ces modules acquièrent ou perdent alors la possibilité de recevoir des connexions dans l'objet où la capsule est insérée.

En revanche comme signalé plus haut, aucune information concernant la surface paramétrique n'est mise en œuvre dans GENESIS 1.5. La définition des surfaces paramétriques suppose en effet, qu'un contrôle de haut niveau des paramètres soit possible, ce dont nous discuterons au chapitre 19. A ce jour, l'édition des paramètres d'une capsule nécessite qu'elle soit ouverte.

16.4 - Conclusion sur les fonctionnalités macro-modulaires

Ensembles et Capsules sont deux notions complémentaires qui répondent, telles qu'elles sont implémentées dans GENESIS 1.5, à une grande diversité de situations de macro-modularité. Les tableaux de la figure 6 montrent comment les enjeux de la macro-modularité identifiés dans le paragraphe 1 de ce chapitre se répartissent sur chacune des deux fonctionnalités.

Enjeux	**Ensembles**	**Capsules**
Faciliter la sélection et la navigation	+ +	
Nommer. Organiser. Communiquer.	+ +	+ +
Associer un traitement mené à plusieurs modules	+ +	
Protéger	+	+ +
Protéger. Réduire la variabilité		+ +
Réduire la charge cognitive	+	+ +
Changer l'échelle du processus de création	+	+ +

	Macro-modularité morphologique	**Macro-modularité fonctionnelle**
Macro-modularité faible	Ensembles.	Ensembles.
Macro-modularité forte	Ensembles protégés. Capsules.	Capsules.

Figure 6 : complémentarité des Ensembles et Capsules.
En haut : répartition des enjeux de la macro-modularité (paragraphe 1 de ce chapitre) en fonction des deux notions Ensemble et Capsules
En bas : répartition des notions d'Ensemble et de Capsule en fonction des types d'ensembles (paragraphe 1 de ce chapitre).

Ensembles et Capsules se distinguent donc avant tout par la *force* des ensembles de modules qu'ils identifient.

Les Ensembles constituent surtout un moyen pratique pour manipuler des parties nommées des objets. Protégés, ils permettent cependant un premier pas vers la macro-modularité forte.

Les Capsules implémentent plus précisément la notion de macro-module. Elles constituent un cadre formel suffisamment puissant pour, à terme, permettre un changement d'échelle dans le processus de création.

Chapitre 17

ANALYSE MODALE ET ACCORDAGE

Nous avons précédemment évoqué le rôle des outils *d'analyse* et de *génération* dans le processus de création. Plusieurs outils de ce type sont encore en cours d'étude. GENESIS propose d'ores et déjà un outil *d'analyse modale* et *d'accordage* dûment testé[1]. Ce chapitre présente ses fondements théoriques, sa mise en œuvre dans GENESIS 1.5 et son ergonomie.

17.1 - Approche théorique de l'analyse modale

L'analyse modale dans GENESIS s'applique au cas des objets topologiques constitués de modules linéaires[2]. Nous considérons dans ce paragraphe l'objet le plus général valide pour cette étude. Il se compose de N modules de type CEL et de N*(N-1)/2 modules de type REF[3].

17.1.1 - Formulation matricielle des algorithmes

Algorithmes des modules

Les CEL de l'objet à analyser seront numérotés de 1 à N. Le CEL i est défini par ses trois paramètres M_i, K_i, Z_i d'inertie, raideur et viscosité dans la base CORDIS [4] – les paramètres K_i et Z_i représentent la raideur et la viscosité externe (ou du milieu). On note :

$X_{i,n}$ sa position à l'échantillon n ;

et avec Δ l'opérateur de dérivation discret (ou de différence).

$\Delta X_{i,n}$ sa vitesse par pas d'échantillonnage, à l'échantillon n ($\Delta X_{i,n} = X_{i,n} - X_{i,n-1}$)

$\Delta^2 X_{i,n}$ son accélération discrète à l'échantillon n ($\Delta^2 X_{i,n} = X_{i,n} - 2*X_{i,n-1} - X_{i,n-2}$) ;

Avec ces notations l'algorithme du CEL i vérifie la relation :

$$M_i \Delta^2 X_{i,n+1} = -K_i X_{i,n} - Z_i \Delta X_{i,n} + F^{\text{int}}_{i,n} + F^{ext}_{i,n} \quad (1)$$

où $F^{\text{int}}_{i,n}$ est la somme des forces exercées par les REF connectés au CEL i ;

et $F^{ext}_{i,n}$ est la somme des forces externes appliquées.

---

[1] Ce chapitre, ainsi que l'outil d'analyse modale de GENESIS, repose pour partie sur des études antérieures menées par Eric Incerti [Incerti.96] et Pirouz Djoharian [Djoharian.90], [Djoharian.93].

[2] Il est possible de procéder à l'analyse modale des objets spatiaux à la condition que les équations qui régissent leur mouvement puissent être approximées par linéarisation pour les faibles niveaux d'excitation. La linéarisation est en général effectuée dans une direction privilégiée de l'espace, et permet d'obtenir un 'objet topologique équivalent' pour certains types de vibrations (par exemple, dans le cas d'une corde, les oscillations transverses – voir l'annexe 1). L'analyse modale consiste alors en l'analyse de cet objet topologique (on parle, dans le cas de la corde, des modes transverses).

[3] Il est toujours possible de substituer à un objet composé de modules MAS, SOL, CEL, RES, FRO, RES et REF un autre objet identique fonctionnellement mais constitué exclusivement de modules CEL et REF : il suffit de remplacer les RES et les FRO par des REF dont certains paramètres sont nuls, et les formules MAT/LIA/SOL par un CEL de raideur et viscosité correspondante. De même, entre deux CEL non connectés, on peut toujours considérer qu'il existe un REF de raideur et viscosité nulle

[4] Voir le Chapitre 10.

Chaque REF de l'objet est repéré par les indices et i et j des CEL qu'il connecte. Le REF (i,j) est défini par ses paramètres K_{ij}, Z_{ij} dans la base CORDIS et vérifie la relation :

$$F_{i \to j,n} = -F_{j \to i,n} = -K_{ij}\left(X_{i,n} - X_{j,n}\right) - Z_{ij}\left(\dot{X}_{i,n} - \dot{X}_{j,n}\right) \quad (2)$$

Combinées, les équations (1) et (2) permettent de calculer pas à pas l'état de l'objet connaissant les forces extérieures qui lui sont appliquées. Il vient pour le CEL i :

$$M_i \ddot{X}_{i,n+1} = -K_i X_{i,n} - Z_i \dot{X}_{i,n} - \sum_{j \neq i} \left[K_{ij}\left(X_{i,n} - X_{j,n}\right) + Z_{ij}\left(\dot{X}_{i,n} - \dot{X}_{j,n}\right)\right] + F_{i,n}^{ext} \quad (3)$$

Formulation matricielle

En écrivant l'équation (3) pour chacun des N CEL de l'objet, on aboutit à la formulation matricielle de l'algorithme de l'ensemble de l'objet :

$$M.\ddot{X}_{n+1} = -KX_n - Z.\dot{X}_n + F_n^{ext} \quad (4)$$

où :

X_n, $\dot{X}_n$ et $\ddot{X}_n$ sont les vecteurs des positions, vitesses et accélérations des CEL ;

M est la matrice carrée diagonale des inerties :

$$m_{ij} = 0 \quad \text{si } j \neq i$$
$$m_{ii} = M_i$$

K est la matrice carrée symétrique des raideurs :

$$k_{ij} = -K_{ij} \quad \text{si } j \neq i$$
$$k_{ii} = K_i + \sum_{k \neq i} K_{ik}$$

Z est la matrice carrée symétrique des viscosités :

$$z_{ij} = -Z_{ij} \quad \text{si } j \neq i$$
$$z_{ii} = Z_i + \sum_{k \neq i} Z_{ik}$$

Système réduit

Nous supposerons dans la suite que toutes les inerties sont strictement positives[1]. En posant :

$$X_n' = M^{1/2} X_n \qquad K' = M^{-1/2} K M^{-1/2} \qquad Z' = M^{-1/2} Z M^{-1/2} \quad (5)$$

avec $M^{1/2} = DIAG\left(\sqrt{M_i}\right)$ et $M^{-1/2} = DIAG\left[1/\sqrt{M_i}\right]$, il est possible de se ramener à une forme matricielle réduite, dans laquelle l'algorithme de l'objet s'écrit :

$$\ddot{X}'_{n+1} = -K'X'_n - Z'.\dot{X}'_n + M^{-1/2} F_n^{ext} \quad (6)$$

Une telle formulation est possible pour *tout* objet CORDIS-ANIMA constitué de modules linéaires. Mettre en œuvre l'analyse modale c'est, partant de ce système réduit, chercher une base orthonormale qui le diagonalise. Les matrices K' et Z' sont symétriques réelles, donc chacune est diagonalisable dans une base orthonormée réelle. Rien n'indique, par contre, que cette base soit commune, de sorte que deux cas se présentent.

[1] Un objet comprenant une inertie nulle diverge, du fait de l'algorithme du module MAS. Il est par contre possible de spécifier dans GENESIS des inerties strictement négatives – mais les objets sont alors très instables. L'analyse modale, cependant, se limite au cas des inerties positives.

17.1.2 - Cas des objets à viscosité compatible

On dira précisément que l'objet est à *viscosité compatible* lorsque K' et Z' sont diagonalisables dans la même base.

Il faut et suffit pour cela que les K' et Z' commutent. Cette propriété, cependant, n'offre guère d'interprétation physique. Elle permet tout au plus de discuter de quelques cas particuliers.

Une condition suffisante – dite de Raileygh – pour que $K'Z'=Z'K'$ est que Z' puisse s'écrire comme un polynôme de K', c'est à dire que :

$$Z'=c_1.Id+c_2.K'+c_3.K'^2... \quad , \; c_1,c_2c_3 \quad . \qquad (6)$$

Rapportée aux matrices M, K et Z de l'objet d'origine et compte tenu de la définition de K' et Z'la condition de Raileygh s'écrit :

$$Z=c_1.M+c_2.K+c_3.KM^{-1}K... \qquad (7)$$

Des deux premiers termes de cette équation on peut déduire quelques configurations, finalement assez courantes, qui assurent que l'objet soit à viscosité compatible :

- Celle où la viscosité est proportionnelle à la raideur pour chaque module CEL *et* REF ;
- Celle où seule la viscosité de milieu (i.e. : la viscosité des CEL) est non nulle et où elle est proportionnelle à l'inertie des CEL ;
- Celle où pour chaque REF la viscosité est proportionnelle à la raideur, où la raideur de milieu est nulle et où la viscosité de milieu est proportionnelle à l'inertie du CEL.
- Etc.

Quoi qu'il en soit, pour tous les objets à viscosité compatible il existe une matrice orthonormée de changement de base Q qui vérifie les trois propriétés :

$${}^tQ=Q^{-1}$$ (orthogonalité)

${}^tQK'Q={}^tQM^{-1/2}KM^{-1/2}Q=K^m$, avec $K^m=\begin{pmatrix} K_1^m & ... & 0 \\ ... & ... & ... \\ 0 & ... & K_N^m \end{pmatrix}$ la matrice diagonale des valeurs propres réelles $(K_1^m,K_2^m,...K_N^m)$ de K', aussi appelées *raideurs modales*

${}^tQZ'Q={}^tQM^{-1/2}ZM^{-1/2}Q=Z^m$, avec $Z^m=\begin{pmatrix} Z_1^m & ... & 0 \\ ... & ... & ... \\ 0 & ... & Z_N^m \end{pmatrix}$ la matrice diagonale des valeurs propres $(Z_1^m,Z_2^m,...Z_N^m)$ de Z', ou *viscosités modales.* (8)

Enfin, la matrice $M^{-1/2}.Q$ joue ici et dans la suite un rôle important. Nous verrons que ses colonnes représentent les *déformées modales* de chaque mode de l'objet. Nous notons $Q^m=M^{-1/2}.Q$ cette *matrice des déformées modales*.

Si on pose maintenant :

$$Y_n = {}^tQX'_n = {}^tQ^m X_n \text{ et } F_n^{ext,m} = {}^tQ^m F_n^{ext} \quad (8)$$

Y et ses dérivées vérifient le *système modal* :

$$\ddot{Y}_n = -K^m . Y_n - Z^m . \dot{Y}_n + F_n^{ext,m} \quad (9)$$

17.1.3 - La chambre modale

Le système (9) est diagonal et se réduit à N équations indépendantes. Il correspond à la formulation matricielle d'un nouvel objet CORDIS-ANIMA, ou *objet modal*, constitué de N oscillateurs physiques *non couplés,* c'est à dire de N modules CEL de paramètres $M=1, K=K_i^m, Z=Z_i^m$ ($0 \le i \le N$).

En adjoignant à l'objet modal la matrice des déformées Q^m, on obtient une « chambre modale » [Djoharian.90][1]. Les équations (4), (5) et (9) montrent que, dans le cas des objets à viscosité compatible, la chambre modale permet une description exhaustive des phénomènes physiques dont l'objet d'origine peut être le siège. Elle peut lui être substituée sans aucune perte d'information (figure 7). On rejoint alors précisément le principe de la synthèse modale[2].

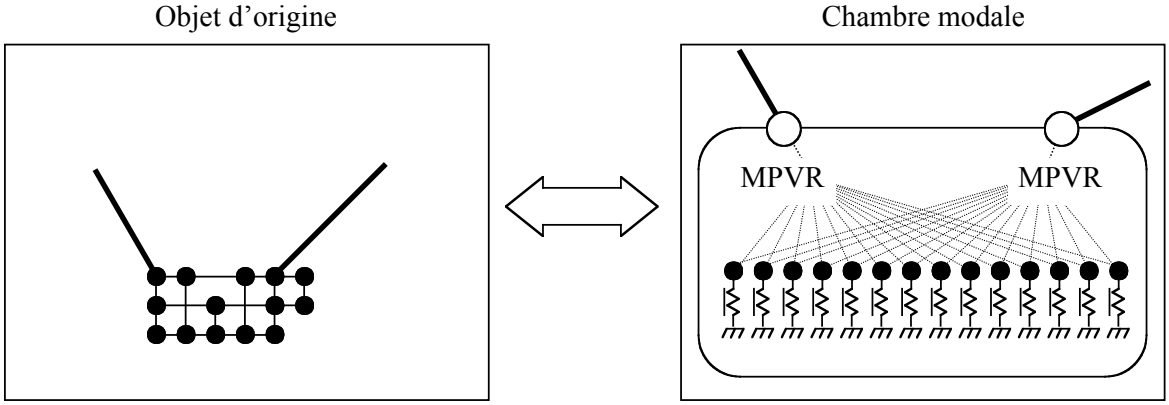


Figure 7 : la chambre modale

La simulation de la chambre modale en lieu et place de l'objet d'origine est possible avec CORDIS-ANIMA sans contredire les principes du formalisme. Il suffit pour cela de mettre œuvre des modules de modification de point de vue relatif MPVR, qui réalisent le passage entre les deux représentations, c'est à dire qui répartissent sur les différents modes les forces entrantes sur chaque CEL conformément à l'équation :

$$F_n^{ext,m} = {}^tQ^m . F_n^{ext} \quad (11)$$

et, une fois les nouvelles positions de l'objet modal calculées, recomposent le déplacement de l'objet d'origine – au minimum au niveau de ses points d'accès – suivant la relation :

$$X_n = Q^m Y_n \quad (12)$$

[1] Djoharian écrit : « La chambre modale « représente la structure vibrante sous forme d'une boîte munie de deux fenêtres d'entrée et de sortie définies par une matrice et sa transposée et contenant une série d'oscilateurs simples indépendants. (…) Les oscillateurs représentent les différents modes caractéristiques de vibration de la structure et les fenêtres représentent les déformées modales associées aux différents modes » [Djoharian.90].

[2] Voir l'Annexe A.

La boucle de simulation prend alors la forme :

- Phase LIA ;
- Phase MPVR 1 : détermination des forces entrantes dans le modèle modal ;
- Phase MAT ;
- Phase MPVR 2 : détermination des positions des CEL d'accès de l'objet d'origine ;

Simuler la chambre modale peut à première vue offrir un gain en termes d'efficacité algorithmique puisque dans l'algorithme de l'objet modal ne figure aucun LIA. Le calcul des modules MPVR nécessaire au changement d'espace de représentation est cependant coûteux. L'optimisation espérée devient donc caduque lorsque que le nombre des points d'accès à l'objet d'origine augmente. C'est notamment le cas dès lors qu'on veut visualiser les mouvements de l'objet et qu'il devient nécessaire de recomposer les déplacements de chacun de ses MAT.

Dans la partie VI, nous avons montré que la visualisation des simulations participe intimement du processus de création. En outre, il ne faut pas perdre de vue que l'équivalence de l'objet et de la chambre modale dépend de la précision des calculs de diagonalisation. Une faible erreur dans ces calculs, inévitable du fait des erreurs numériques (précision de l'algorithme de diagonalisation et erreurs de quantification et d'arrondi), risque d'induire une divergence progressive entre les phénomènes résultant de la simulation des deux modèles. Dans le contexte de GENESIS, c'est le modèle topologique qui constitue la référence, de sorte qu'il nous a semblé préférable de ne jamais avoir recours à la chambre modale pour réaliser les simulations. C'est ailleurs que se trouve l'intérêt de l'analyse et de la représentation modales.

17.1.4 - Enseignements de la représentation modale

Les enseignements qu'apporte la représentation modale des objets CORDIS-ANIMA sont comparables à ceux que propose l'analyse modale des objets réels, bien connue du physicien. Nous en préciserons quelques-uns.

Oscillation libres

Dans le cas des oscillations libres ($F_n^{ext}=0$), le mouvement de l'objet ne dépend que de ses conditions initiales X_0, X_1, qui se décomposent sur le système modal en

$$Y_0={}^tQ^m X_0 \text{ et } Y_1={}^tQ^m X_1$$

Dès lors que son amortissement n'est pas critique, chaque oscillateur élémentaire j de l'objet modal suit un mouvement pseudo-périodique de la forme :

$$Y_{j,n} = \alpha_j . e^{-\lambda_j^n \cdot \frac{n}{Fe}} \cos\left(2\pi f_j^n \cdot \frac{n}{Fe} + \varphi_j\right) \quad (13)$$

avec :

α_j et φ_j deux constantes dépendant des conditions initiales choisies

f_j^m la fréquence propre et λ_j^m le coefficient d'amortissement [1] de l'oscillateur numérique, dont les valeurs sont connue[1] :

[1] On peut également considérer le *temps d'amortissement* T_j^m (temps nécessaire pour que l'amplitude soit réduite à (1/e) soit environ 35% de sa valeur initiale).

$$f_j^m = \frac{F_e}{2\pi}.\mathrm{Arccos}\left[\frac{2\left(K_j^m + Z_j^m\right)}{2\sqrt{1-Z_j^m}}\right] \qquad (14)$$

$$\sigma_j^m = -\frac{F_e}{2}.\ln\left(1-Z_j^n\right) \qquad (15)$$

Puisque $X_n = Q^m Y_n$, il vient que les mouvements des CEL de l'objet d'origine s'écrivent chacun comme une combinaison linéaire de termes pseudo périodiques. Pour le CEL i, par exemple :

$$X_{i,n} = \sum_j \left[Q_{i,j}^m * \alpha_j . e^{-\sigma_j^n . \frac{n}{Fe}} . \cos\left(2\pi f_j^n . \frac{n}{Fe} + \varphi_j\right)\right] \qquad (16)$$

De cette équation (16) on peut déduire deux informations intéressantes quant aux propriétés de l'objet d'origine :

- Tout d'abord, les fréquences caractéristiques de ses mouvements sont bien sûr les fréquences propres des différents modes de l'objet modal. Dans l'hypothèse où, par exemple, l'objet est une structure vibrante, ces fréquences seront constitutives du spectre des sons qu'il génère.
- Ensuite, le terme $Q_{i,j}^m$ exprime la contribution du mode j au déplacement du CEL i de l'objet d'origine. C'est donc bien que les colonnes de la matrice Q^m représentent les *déformées modales* de chaque mode. Si $Q_{i,j}^m$ est nul, le CEL i est un *nœud* pour le mode j. S'il est maximal, c'est un ventre. La connaissance de la matrice Q^m permet de décider où exciter et observer l'objet pour que certains modes, et donc certaines fréquences, soient privilégiés ou au contraire atténués.

Oscillations forcées ; réponse en fréquence

Pour étudier la réponse en fréquence de l'objet d'origine entre le CEL i (disons : son point d'accès) et le CEL k (disons : son point d'observation ou , si l'objet est une structure vibrante, son point d'écoute), il est pratique de considérer qu'en i une force pulsante *complexe* de fréquence donnée $f_\omega = F.e^{j\omega n}$ est appliquée.

Comme le montre l'équation (11), la colonne i de la matrice ${}^tQ^m$ représente la sensibilité des N modes à une excitation au niveau du CEL i. La force f_ω se distribue sur l'ensemble des modes j suivant :

$$f_j^m = {}^tQ_{i,j}^m * f_\omega \qquad (17)$$

Le mode j est donc lui aussi soumis à une force pulsante de même fréquence et de même phase, mais d'amplitude ${}^tQ_{i,j}^m$.

[1] D'après [Incerti.96 pp55-60]. Notons que le temps d'amortissement s'écrit : $T_j^m = \frac{2}{F_e} * \frac{0.1}{\ln\left(1-Z_j^m\right)}$.

La *fonction de transfert* complexe $H_j(\omega)$ de l'oscillateur physique j peut être déterminée algébriquement[1]. En l'utilisant, on peut écrire le mouvement du mode j sous la forme :

$$Y_{j,n} = {}^{t}Q^{m}_{i,j}.H_j(j\omega)Fe^{j\omega n} \quad (18)$$

Il nous faut maintenant 'remonter' à l'objet d'origine. Puisque $X = M^{-1/2}QY$ le mouvement du CEL k que nous voulons observer suit :

$$X_{k,n} = \sum_j \left[Q^{m}_{kj}.Q^{m}_{i,j}.H_j(j\omega). \right] * Fe^{j\omega n} \quad (19)$$

Cette équation montre que la connaissance du système modal permet de déterminer algébriquement la *fonction de transfert* (ou réponse harmonique) de l'objet d'origine entre deux des CEL qui le constituent.

17.1.5 - Utilisation du système modal ; accordage

Ajustement individuel des modes

Disposant d'un *objet modal* qui répond au formalisme CORDIS-ANIMA, il est naturel de vouloir le modifier. Qui plus est, puisque les équations (14) et (15) offrent une relation entre la fréquence propre et le temps d'amortissement des modes d'une part, leurs paramètres K^{m}_{i}, Z^{m}_{i} d'autre part, on peut espérer pouvoir régler indépendamment les constantes de temps de chaque mode de l'objet.

Cet *accordage mode à mode*, cependant, n'est pas sans poser problème.

En effet, dès lors que les matrices diagonales des raideurs modales K^{m} et des viscosités modales Z^{m} sont modifiées en K^{m}_{2} et Z^{m}_{2}, il n'est pas garanti que le retour à la représentation initiale par les matrices de passage amène un réseau de connexion identique à celui de l'objet d'origine. Il se peut que les matrices K et $K_2 = M^{1/2}Q.K^{m}_{2}.{}^{t}Q.M^{1/2}$ (resp. les matrices Z et $Z_2 = M^{1/2}Q.Z^{m}_{2}.{}^{t}Q.M^{1/2}$) n'aient plus les mêmes éléments nuls, c'est à dire que l'opération d'accordage mode à mode implique la création ou la suppression de modules LIA dans l'objet d'origine.

Or, il est certain que le réseau de connexion MAT/LIA, est plus que toute autre chose représentatif de ce qu'est l'objet pour l'utilisateur. La modification de structure que peut induire l'accordage mode à mode risque de déstabiliser l'utilisateur, et à terme de nuire à sa représentation mentale des objets, et plus généralement de CORDIS-ANIMA. Que penser, en effet, d'un objet qui modélise une corde mais dans lequel, suite à l'accordage individuel des modes de l'objet modal qui lui correspond, chaque MAT se retrouverait connecté à l'ensemble des autres ? Que penser, à l'inverse, d'un objet qui modélise une membrane mais dans lequel le réseau de connexion serait celui d'un filament ?

[1] Nous ne détaillons pas l'expression de la fonction de transfert $H_j(j)$, relativement complexe dans le cas de l'oscillateur discret et de peu d'intérêt ici – voir [Incerti.96 pp57-59]. Rappelons toutefois que la fréquence de résonance du mode j est toujours légèrement inférieure à sa fréquence propre.

En outre, nous considérons, incidemment, que la dualité existant entre la représentation modale et la représentation 'spatiale' (topologique, dans le cas de GENESIS), acquise de longue date par les physiciens, est difficile à aborder pour l'utilisateur de GENESIS. En substituant à une 'pensée physique' intuitive une approche tournée vers les propriétés fréquentielles et temporelles des objets, le principe de l'accordage mode à mode introduirait un positionnement fort éloigné de celui que nous souhaitons promouvoir.

Nous n'avons pas pour l'heure autorisé, dans GENESIS, un accordage individuel des modes. L'introduction d'une telle fonctionnalité nécessite, compte tenu des remarques précédentes, une réflexion approfondie qui devra se poursuivre[1].

Accordage

Il est possible, en revanche, de proposer une approche intermédiaire n'autorisant qu'une modification globale des paramètres physiques K_i^m, Z_i^m des modes à travers deux coefficients multiplicateurs α_K, α_Z, de sorte que :

$$K_2^m = \alpha_K . K^m \text{ et } Z_2^m = \alpha_Z . Z^m \qquad (20)$$

Les équations (8) montrent que pour opérer une telle modification sur l'objet modal il suffit de multiplier toutes les raideurs et viscosités de l'objet d'origine par les mêmes coefficients α_K et α_Z, opération qui n'entraîne aucune modification de la structure du réseau.

Avec les équations (14) et (15) qui lient les fréquences et les temps d'amortissement de chaque mode à ses paramètres physiques K_i^m, Z_i^m, il est possible de déterminer α_K et α_Z de telle sorte que l'un des modes de l'objet, choisi arbitrairement (par exemple le mode le plus bas), ait la fréquence et le temps d'amortissement voulus. Les autres modes, cependant, sont alors également affectés. Pour peu que α_K et α_Z soient suffisamment petits devant 1, un développement limité de l'équation (14) montre que l'ensemble du spectre sera décalé en fréquence, les rapports entre les fréquences des différents modes étant cependant conservés. On aura donc effectué un *accordage global* de l'objet d'origine, dont l'effet sera comparable à celui que produit, par exemple, le violoniste lorsqu'il module la tension des cordes de son instrument.

[1] De fait, les nombreuses utilisations de GENESIS n'ont pas jusqu'à aujourd'hui montré qu'un contrôle précis et individualisé des constantes de temps des modes soit effectivement nécessaire. En outre, plutôt que l'accordage mode à mode d'un objet topologique dont on dispose de la représentation modale, il nous semblerait plus intéressant d'introduire dans GENESIS des principes relevant directement de la synthèse modale en implémentant les modules MPVR et en autorisant un travail direct sur des 'objets modaux' – nous y reviendrons dans le chapitre 19.

17.1.6 - Cas des objets à viscosité non compatible

L'étude menée jusqu'à ce point concerne le cas bien particulier des objets à viscosité compatible. Les choses se compliquent dans le cas général, lorsque K' et Z' ne sont pas diagonalisables dans une même base. Deux approches sont cependant possibles.

Etude du système de taille 2N ; modes complexes

On peut tout d'abord s'intéresser au système de taille 2N qui lie le vecteur $\Box_n = \begin{pmatrix} X_n \\ \dot{X}_n \end{pmatrix}$ et sa dérivée discrête $\dot{\Box}_n = \begin{pmatrix} \dot{X}_n \\ \ddot{X}_n \end{pmatrix}$, système qui compte tenu de l'équation (5) s'écrit :

$$\dot{\Box}_n = \begin{pmatrix} 0 & Id_N \\ -K' & -Z' \end{pmatrix} \Box_n + \begin{pmatrix} F'_n \\ 0 \end{pmatrix} \qquad (21)$$

Il suffit d'étudier la matrice ainsi obtenue pour avoir toutes les informations requises sur le système original.

Par contre, cette matrice n'est pas symétrique et donc n'est pas forcément diagonalisable. A supposer qu'elle le soit, rien n'indique en outre que ses valeurs propres soient réelles. On obtiendra donc deux matrices de passage complexes *non orthogonales* de taille 2N, ainsi que 2N modes *complexes* conjugués deux à deux[1].

Puisque Q n'est plus orthogonale et devient complexe, l'analyse du système 2N n'offre plus une notion de déformée modale aussi claire que précédemment. Les N premières lignes de la matrice Q^{-1} expriment toujours l'importance du mode j au niveau du CEL i, mais lorsqu'un mode unique est excité le déphasage entre les CEL de l'objet d'origine n'est plus égal à 0 ou π. Les N premières lignes de la matrice Q expriment la façon dont les mouvements des CEL se répartissent sur les différents modes ; cependant ${}^tQ$ n'est plus égal à Q^{-1} : la répartition de l'objet d'origine sur les différents modes n'est plus symétrique de la répartition des modes sur l'objet d'origine.

Un autre problème plus pragmatique posé par l'approche, en supposant toujours que la matrice soit diagonalisable, est qu'il faut alors diagonaliser une matrice de taille 2N, qui plus est non symétrique, puis inverser la matrice de passage Q qui n'est plus orthogonale. Dans le cas général, cela peut être très coûteux[2]. Or, l'analyse modale n'est qu'un outil périphérique pour l'utilisateur de GENESIS, qui n'accepterait pas de patienter un temps considérable dans l'attente de ses résultats. Il serait donc préférable d'éviter la manipulation du système 2N et les nombreux problèmes qu'elle pose[3]. Fort heureusement, une approximation est très souvent possible.

---

[1] Puisque la matrice d'origine est réelle.

[2] La calculs à mener sur le système 2N sont bien plus complexes que ceux évoqués précédemment dans le cas des matrices de tailles N symétriques, non seulement du fait du doublement de la taille mais encore du fait que les algorithmes efficaces dans le cas des matrices symétriques ne sont plus valables. Plusieurs algorithmes existent cependant. Les algorithmes généraux, tel la méthode de la puissance, deviennent extrêmement lourds dès lors que la taille 2N de la matrice atteint quelques dizaines. Les autres algorithmes tels la méthode 'QR' ne sont envisageables que pour certains 'conditionnements' (critère assez vague et propre à chaque algorithme selon lequel une matrice se prête bien ou mal au traitement envisagé).

[3] Le calcul des modes complexes constitue, à long terme, une possibilité de développement de GENESIS. De nombreuses autres fonctionnalités sont plus essentielles à court terme.

Etude par approximation

Dans cette seconde approche, seule la matrice symétrique des raideurs K' est diagonalisée dans une base orthonormée réelle. L'objet étudié est donc équivalent à celui que définit le système :

$$\ddot{Y}_n + \Omega_{K'}^2 Y_n + {}^tQ^m Z Q^m \dot{Y} + {}^tQ^m F_n^{ext} \qquad (22)$$

La matrice ${}^tQ^m.Z.Q^m$, cependant, n'est plus diagonale, et ce système ne peut guère être envisagé tel quel. C'est ici qu'on peut décider de simplifier considérablement le problème, en ne considérant que les éléments diagonaux de ${}^tQ^m.Z.Q^m$.

Le système 'pseudo-modal' :

$$\ddot{Z}_n + \Omega_{K'}^2 Z_n + DIAG\left({}^tQ^m.Z.Q^m\right)\dot{Z} + {}^tQ^m F_n^{ext} \qquad (23)$$

ainsi constitué est similaire au système (9) obtenu dans le cas des objets à viscosité compatible, et permet toutes les considérations évoquées plus haut. Il n'est plus, par contre, parfaitement équivalent à l'objet d'origine – et il convient de s'interroger sur l'erreur introduite.

Il faut remarquer tout d'abord qu'il deviendrait risqué, voire impossible de simuler la chambre pseudo-modale ainsi définie en lieu et place de l'objet d'origine, puisque l'erreur introduite s'accumule progressivement pendant la simulation et induit finalement une différence importante entre les phénomènes générés et ceux que génère la simulation 'directe'.

Cependant, dans une perspective d'analyse, l'approximation est très souvent satisfaisante. Djoharian a montré qu'on obtient alors une bonne approximation des modes dès lors que les valeurs propres de Z sont petites devant celles de K [Djoharian.97][1].

En posant :

$$\varepsilon = \frac{\max_j\left(\left|\omega_j^K\right|\right)}{\max_j\left(\left|\omega_j^Z\right|\right)} \qquad (24)$$

avec ω_j^K et ω_j^Z les valeurs propres de K et Z respectivement, on peut écrire :

$$\omega_j^Z = \left({}^tQ^m Z Q^m\right)_{j,j} + o(\varepsilon)$$

[1] Bien qu'il soit possible de reporter cette condition sur les matrices K et Z originelles, la formulation ainsi obtenue est quelque peu absconse. On retiendra que l'approximation est valable quand les valeurs des viscosités sont petites devant celle des raideurs. C'est, le plus souvent, le cas dans les objets CORDIS-ANIMA.

17.2 - L'analyse modale dans GENESIS

17.2.1 - L'utilitaire AnModTop

L'analyse modale est rarement effectuée sur des objets complets, mais plutôt sur un groupe fonctionnel de modules : une structure vibrante, un chef, un instrumentiste par exemple. On peut donc considérer que la taille maximale des objets à analyser n'excède jamais N=1000 MAT, de sorte que la taille des matrices à diagonaliser, inférieure à 1000x1000, reste raisonnable. Cependant, quelle que soit la technique numérique utilisée, la diagonalisation d'une matrice n'est pas un calcul simple. Il n'est pas envisageable qu'il interdise l'interaction de l'utilisateur avec GENESIS, et donc ne saurait être effectué au sein du programme GENESIS lui-même sans nuire considérablement à son utilisabilité. Il doit être déporté dans un utilitaire. C'est le rôle du programme AnModTop. Cet utilitaire résulte d'une adaptation du programme PigMag conçu en 1999 par Pirouz Djoharian, qu'il a fallu développer pour l'intégrer à l'environnement GENESIS.

Diagonalisation et extraction des données modales

Les techniques numériques de diagonalisation sont nombreuses. L'algorithme de Jacobi, dédié au cas des matrices symétriques qui nous préoccupe, est un choix intéressant : il est fiable et stable, et surtout permet d'obtenir à la fois l'Ensemble des valeurs propres de la matrice à diagonaliser (ici : les raideurs modales) et la base de diagonalisation (ici : les déformées modales). C'est lui qui est au cœur du programme AnModTop et de son prédécesseur PigMag[1].

L'algorithme de Jacobi est un algorithme itératif qui converge progressivement. AnModTop définit une mesure de la 'diagonalité' de la matrice par :

$$MESURE_k = \frac{1}{N^2} \cdot \frac{\sum_{i \neq j} \left(K^k_{i,j} * K^k_{i,j} \right)}{\sum_{i \in [1..N]} K^k_{i,i}} \quad (26)$$

L'algorithme est stoppé lorsque MESURE devient inférieur à une 'finesse', qui vaut 10^{-12} par défaut, et qui le cas échéant est paramétrable.

Il est important que l'utilisateur puisse être informé régulièrement de la progression du calcul. Il faut, pour cela, être capable de mesurer le taux de convergence de l'algorithme de diagonalisation. Dans AnModTop, le taux de convergence est mesuré après chaque itération par la valeur :

$$TAUX_k = \frac{\log(MESURE_k) - \log(MESURE_0)}{\log(finesse) - \log(MESURE_0)} * 100$$

La valeur de TAUX évolue entre 0 et 100. La figure 8 donne un exemple de sa croissance. La forme générale de la courbe est toujours la même quel que soit l'objet analysé. Toutefois nous n'avons pas trouvé de fonction d'interpolation valable dans tous les cas.

[1] L'algorithme de Jacobi bien connu des ingénieurs repose sur une transformation pas à pas de la matrice à diagonaliser par des rotations orthogonales successives qui annulent progressivement les termes non diagonaux. D'autres algorithmes a sont particulièrement efficaces sont moins complet, ne permettant d'obtenir qu'un couple valeur propre/vecteur de diagonalisation à la fois (cas de l'algorithme de Lanczos), ou encore que les seules valeurs propres (méthode de tridiagonalisation et bissection). Un exposé détaillé de ces divers algorithmes, et particulièrement de l'algorithme de Jacobi pourra être trouvé dans [Lascaux.86].

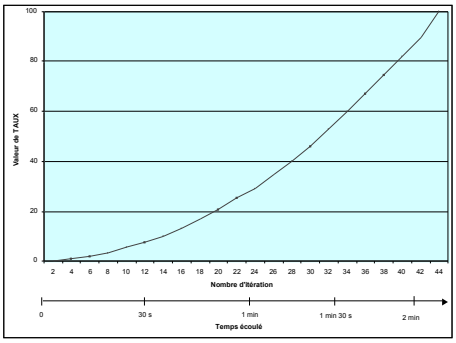


Figure 8 : évolution de TAUX en fonction du nombre d'itération et du temps.
*Le modèle ici analysé est une membrane carrée (20*20 CEL, 760 REF).*

L'exécution de AnModTop fait appel à 5 phases successives :

- 1 – Le chargement d'un objet enregistré au format .top, dont le nom est passé en ligne de commande. Seuls sont acceptés les objets composés exclusivement de modules MAS, SOL, CEL, RES, FRO, REF et pour dans lesquels toutes les inerties sont positives. Dans le cas contraire, AnModTop génère une erreur.
- 2 - Une phase de pré-conditionnement qui construit à partir de l'objet d'origine sa représentation matricielle réduite, conformément à l'équation (5). Les matrices étant symétriques, il suffit de considérer (N*(N-1))/2 coefficients.
- 3 - La diagonalisation de la matrice K' par l'algorithme de Jacobi et ce faisant l'élaboration progressive de la matrice de passage Q.
- 4 - Une phase de post-conditionnement qui, connaissant Q, calcule les valeurs les viscosités modales approchées (On a noté que ces viscosités modales ne sont exactes que pour les objets à viscosité compatible) et calcule la matrice des déformées Q^m .
- 5 – L'écriture du fichier contenant les données modales.

Les fichiers .ana

Lorsque l'analyse est terminée, un fichier .ana est créé. Le format des fichiers .ana est simple. Y figurent le nombre de modes de l'objet (identique au nombre de MAS et de CEL) et, pour chaque mode j classé par ordre de raideur croissante :

- La raideur modale K_j^m
- La viscosité modale Z_j^m
- La déformée modale, c'est à dire la colonne j de la matrice Q^m

Ainsi, le format des fichiers .ana privilégie une représentation physique des modes à une représentation temporelle (fréquence propre et temps d'amortissement de chaque mode). La représentation physique des modes est en effet de plus bas niveau. Elle garantit la possibilité d'un nombre maximum de traitements ultérieurs (dont l'accordage) sur les données modales.

Communication avec GENESIS

L'intégration d'AnModTop dans l'environnement GENESIS repose sur une communication par messages permettant d'informer le programme père de l'état d'avancement de l'analyse. La figure 9 résume les échanges de données entre GENESIS et les programmes AnModTop.

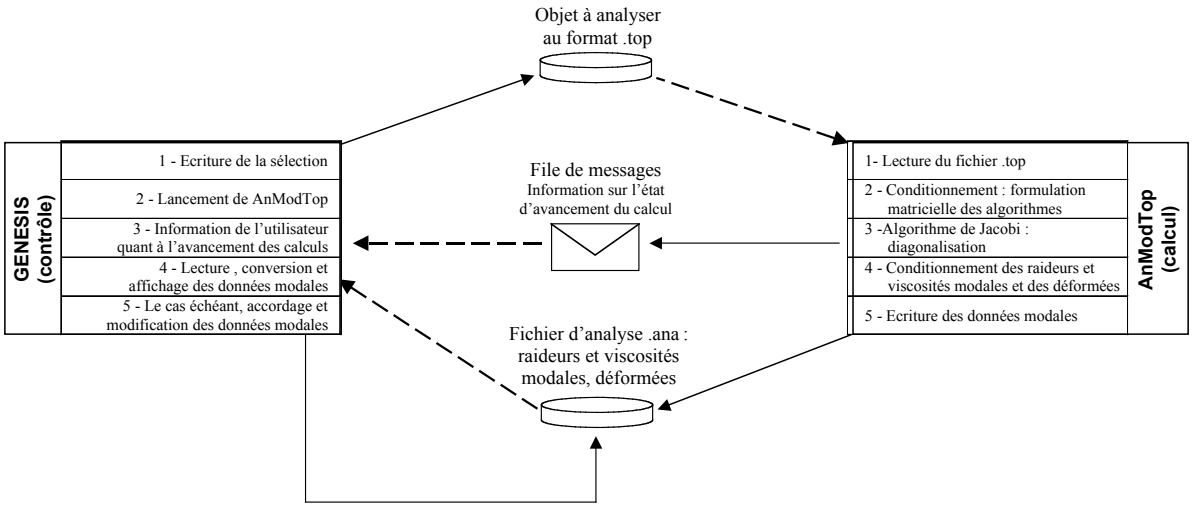


Figure 9 : Communication entre GENESIS et le programme AnModTop

17.2.2 - Ergonomie

Articulation entre Analyse et Ensemble

Dans GENESIS, une analyse modale est toujours relative à un Ensemble. Lorsqu'une analyse est lancée sur une sélection un Ensemble englobant cette sélection est généré – si l'utilisateur n'en a pas créé un auparavant. Toute modification ultérieure du contenu de l'Ensemble ainsi constitué provoque l'affichage d'un message de confirmation, indiquant que les données d'analyse vont être perdues. Pour afficher le résultat d'une analyse, il faut et suffit de sélectionner le contenu de l'Ensemble correspondant sur l'établi – ou dans la fenêtre de gestion des Ensembles.

L'ergonomie mise en œuvre pour afficher les résultats de l'analyse repose sur deux fenêtres. A chacune de ces fenêtres correspond un niveau de détail des informations rendues disponibles à l'utilisateur. La seconde est bloquante et affiche toutes les informations du fichier .ana de différentes manières.

La fenêtre simple

La première fenêtre est libre ; elle affiche à tout instant, les informations d'analyse relatives à la sélection réalisée. Elle peut, en conséquence, prendre trois états.

- Lorsque aucune analyse n'a eu lieu ou n'est en cours sur la sélection, il est possible soit de lancer une nouvelle analyse, soit de sélectionner les Ensembles de modules qui ont déjà été analysés. Lorsque l'utilisateur demande une nouvelle analyse, GENESIS effectue une étude préalable de la sélection et, le cas échéant, propose de la modifier pour la rendre analysable – par exemple si elle comprend des modules LNL, ou si certains LIA sélectionnés sont liés à des MAT non sélectionnés. Un Ensemble est ensuite créé.
- Lorsque la sélection courante est en cours d'analyse, la fenêtre affiche l'état d'avancement des calculs dans une barre de progression.

- Lorsque la sélection a été analysée, des informations élémentaires quant à son résultat sont affichées. Si l'analyse s'est déroulée normalement et que plusieurs modes ont été identifiés les informations relatives au premier mode dont la fréquence propre est non nulle – le mode fondamental dans le cas ou l'objet est harmonique – sont affichées (figure 10). Ici, les données physiques contenues dans le fichier .ana sont converties en fréquence et temps d'amortissement – et le cas échéant en note. Nous considérons, en effet, que ce sont ces informations que l'utilisateur recherche avant toute chose lorsqu'il procède à l'analyse d'un objet.

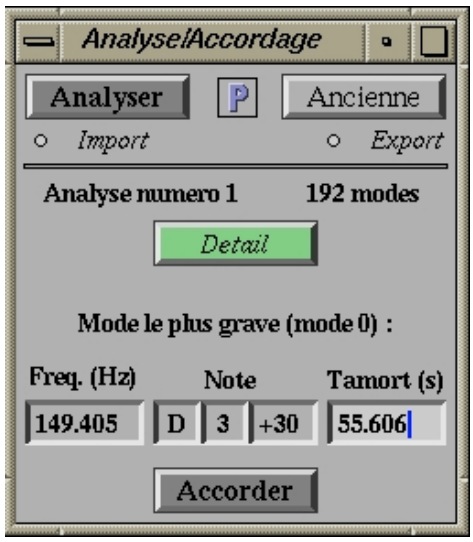


Figure 10 : la fenêtre simple d'analyse (la sélection a été analysée sans erreur) *(couleur p. 447).*

Enfin, la fenêtre simple permet *d'accorder* l'objet analysé, à la fois en termes de fréquence et de temps d'amortissement, de telle sorte que le mode fondamental affiché acquière les propriétés désirées.

La fenêtre de détail

La fenêtre de *détail* (figure 11) répond à deux objectifs :

- Permettre l'accordage global de la sélection sur n'importe quel mode.
- Donner un maximum d'information sur les résultats de l'analyse modale.

L'ergonomie retenue s'organise autour de quatre espaces :

- En haut à gauche, une liste détaillée des modes fait apparaître pour chacun d'eux la fréquence et la période (cette dernière est intéressante lorsque l'objet analysé est un instrumentiste ou un chef), le temps et le coefficient d'amortissement, la raideur et la viscosité modale, l'amplitude en dB – un point d'accès et un point d'observation étant précisés[1].
- En dessous, un espace de saisie permet d'accorder l'objet sur le mode préalablement choisi dans la liste. L'accordage est appliqué immédiatement à l'ensemble des modes.
- A droite, une fenêtre de dessin, dans laquelle apparaît à l'aide d'un dégradé de couleurs la déformée du mode sélectionné[2]. En cliquant sur l'un des MAT représentés dans cette fenêtre, l'utilisateur peut modifier les points d'accès et d'observation considérés pour le calcul des amplitudes des différents modes.

[1] Un point d'accès et un point d'observation sont proposés par défaut en étudiant les connexions de l'objet analysé à son environnement. Par exemple, si un BUT est lié à l'objet analysé, GENESIS suppose que c'est un point d'excitation. L'utilisateur peut, cependant, spécifier un point d'accès et d'écoute différents en agissant sur la déformée modale.

[2] Une représentation tridimensionnelle de la déformée, bien plus attractive pour l'œil, est envisageable. Elle n'apporterait pas cependant d'information plus précise et induirait plusieurs problèmes ergonomiques quant au positionnement du point de vue.

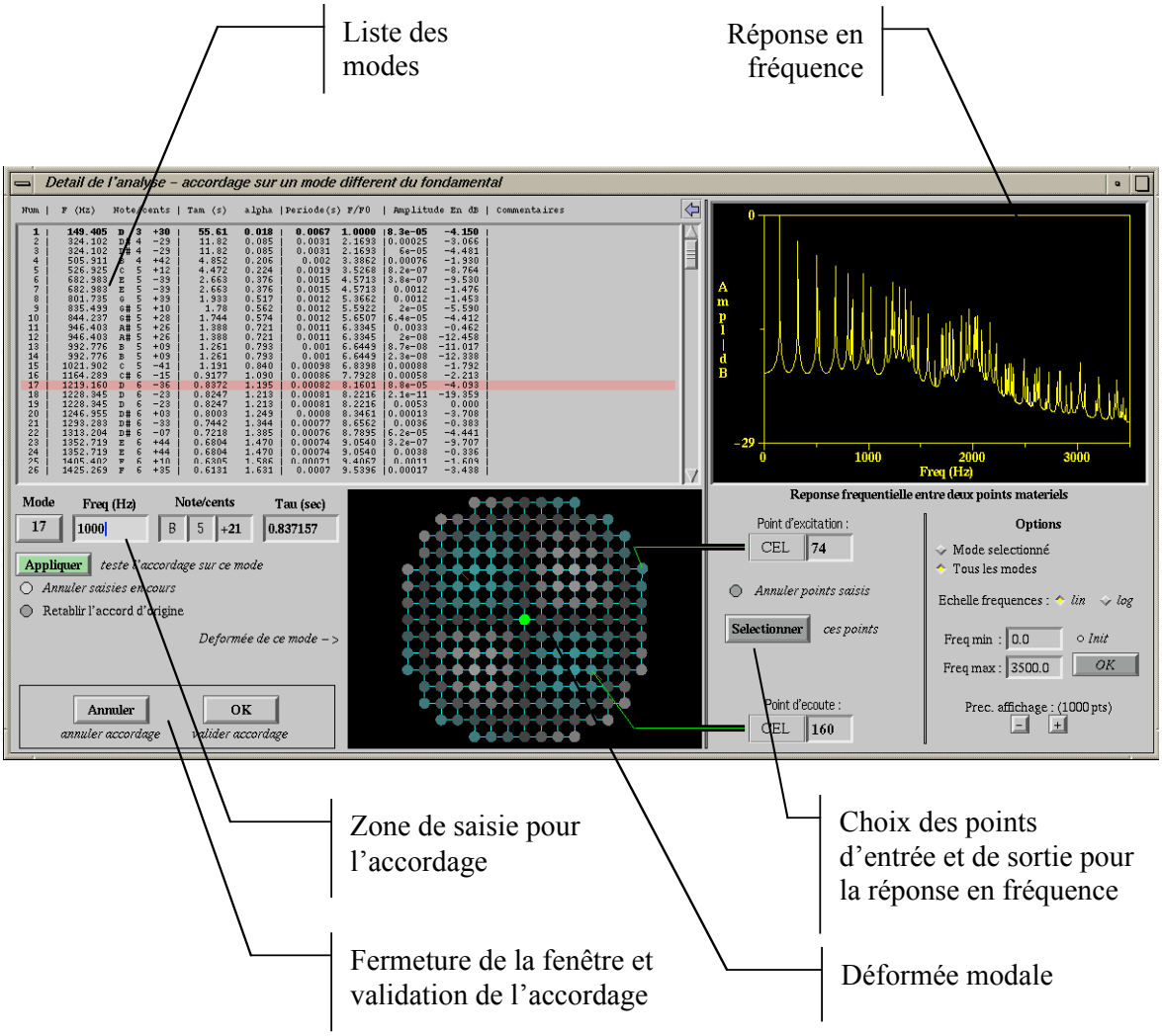


Figure 11 : la fenêtre de détail de l'analyse modale *(Couleur p. 447).*

- A droite enfin, un graphique dans lequel figure la réponse fréquentielle de l'objet, compte tenu des points d'accès et d'observation choisis.

Chacun de ces espaces est actualisé en permanence en fonction des actions de l'utilisateur.

La fenêtre de détail des analyses modales est complète en ce qu'elle offre l'ensemble des fonctionnalités envisageables, mais nous considérons que celles-ci ne sont pas encore parfaitement organisées. Son ergonomie constitue précisément une *mise à plat*[1] : elle a été conçue dans le but de bénéficier de retours utilisateurs pour que puisse être proposée, dans un second temps, une ergonomie optimisée.

[1] Voir la méthodologie de conception de GENESIS, partie II Chapitre 5.

Chapitre 18

OUTILS COMPLEMENTAIRES D'EDITION

Ce troisième chapitre offre un rapide exposé des divers outils qui complètent le mode lutherie de GENESIS 1.5. Ces outils sont simples à la fois conceptuellement et fonctionnellement, mais n'en sont pas moins indispensables pour soutenir le processus de création et faciliter le travail de l'utilisateur. Leur ergonomie a donc été pensée pour être claire et efficace. Nous évoquerons successivement :

- L'outil de génération de structures régulières
- L'outil de substitution de modules
- L'outil de multiplication des paramètres
-

18.1 - Génération de structures régulières

Quelle que soit l'approche de la modélisation plusieurs types de réseaux topologiques apparaissent de façon récurrente dans les objets CORDIS-ANIMA. Lorsque par exemple l'utilisateur cherche des metaphores de corps sonores réels, il est conduit dans un premier temps à s'intéresser aux cordes, membranes, plaques qui sont particulièrement courantes dans l'univers instrumental traditionnel. Leurs pendants CORDIS-ANIMA sont donc souvent mis en œuvre dans GENESIS. Lorsqu'il adopte à l'inverse une démarche exploratoire, il s'intéresse naturellement en premier lieu aux structures régulières avant d'envisager des topologies plus complexes.

L'outil de *génération de structures régulières* a pour objet d'éviter de reconstruire ces divers réseaux topologiques courants.

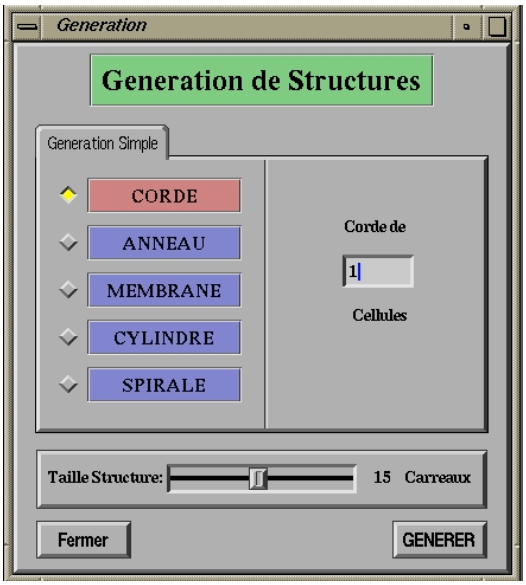


Figure 12 : l'outil de génération de structures régulières

Quatre types de réseaux ont été retenus pour cet outil : la corde, la membrane, l'anneau et la spirale (figure 13). Les deux premiers sont particulièrement utiles en ce qu'ils permettent d'exprimer des metaphores simples de structures vibrantes réelles, telles les cordes, battons, peaux, plaques, cloche, etc. La spirale et l'anneau sont conçus pour faciliter une démarche exploratoire de l'utilisateur.

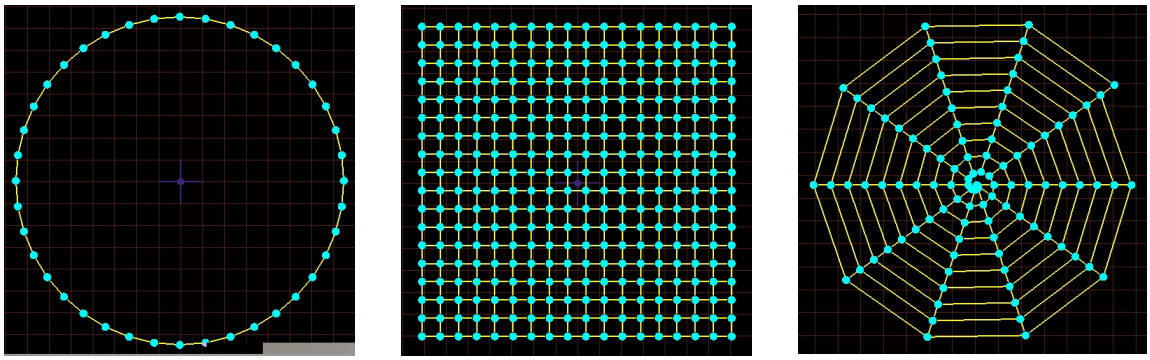


Figure 13 : quelques structures générées. Anneau, membranes, spirale.
*(Couleur p. 445)***.**

La génération de structures régulières n'est pas considérée comme une fin. Au contraire, toute génération est suivie d'une phase d'adaptation, qui permet de converger vers la structure souhaitée en substituant ou supprimant certains modules et en éditant les paramètres (figure 14).

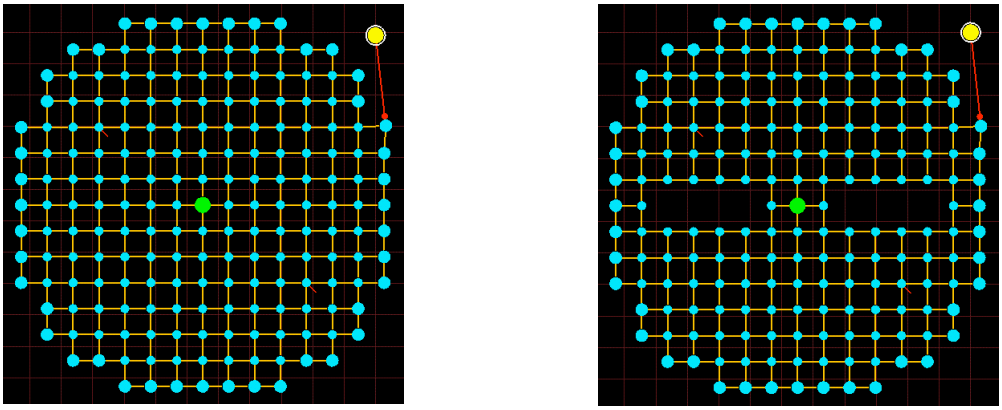


Figure 14 *(Couleur p. 445)* ***: la génération est suivie d'une adaptation***
A gauche, la membrane préalablement générée a été transformée en metaphore de cloche par l'adjonction d'un point fixe central, l'élection de paramètres évoquant le metal, et le choix d'inerties plus importantes au bord.
Dans le modèle de droite, certains modules MAS ont de plus été supprimés, ce qui est à l'image de ce que fait un accordeur lorsqu'il 'gratte' l'intérieur de la cloche. Le spectre des modes de l'objet est ainsi modifié sans que la signature du timbre ne soit affectée.

Nous considérons par ailleurs que la génération de structures régulières n'est qu'une première approche du principe de génération de structure. GENESIS devra, dans les prochaines années, être complété par d'autres procédés de génération. C'est la raison pour laquelle la fenêtre de génération fait apparaître une structure à onglet, qui sera à l'avenir développée. Parmi les développements possibles citons :

- La génération par produit de deux structures[1]
- La génération de structures paramétrées, basée sur l'analyse d'un phénomène existant (un son, par exemple) dans l'objectif d'une re-synthèse.

18.2 - Substitution de modules

La substitution de modules a deux raisons d'être essentielles :

- Elle complète avantageusement la fonctionnalité de génération de structures. Une structure générée est constituée de modules CEL et REF ; elle est couramment suivie d'une substitution afin que le type de modules souhaités puisse être choisi.
- Elle permet de complexifier par étapes les modules constituant un objet et ainsi d'explorer progressivement les propriétés de la matière (figure 15).

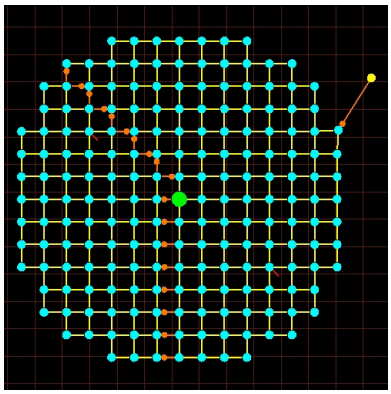


Figure 15 : complexification progressive des propriétés physiques d'un objet par substitution de modules.
Ici, quelques modules REF ont été substitués (voir ci dessous) par des BUT, de sorte que la membrane est 'fêlée'. En choisissant le nombre de REF substitués, il est possible de contrôler l'impact de la fêlure sur le son.

Dans GENESIS 1.5, toute substitution est réalisée de manière à ce que les propriétés physiques du module substitué soient conservées aussi loin que possible (figure 16).

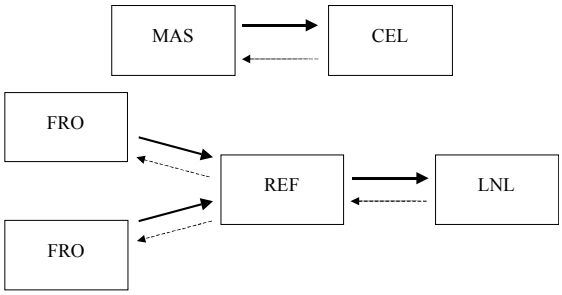


Figure 16 : substitutions
Les flèches continues représentent les substitutions qui conservent les propriétés physiques du module substitué. Une substitution suivant les pointillés peut également être indifférente lorsque le module substitué a certains paramètres nuls.

[1] Voir [Incerti.96].

La figure 17 permet de comparer l'apparence de la fenêtre de substitution de GENESIS 1.5, à celle de la version 1.3. L'ensemble constitue un bon exemple d'évolution résultant de la démarche itérative mise en œuvre pendant nos travaux en montrant comment une ergonomie chargée peut être remplacée avantageusement par une ergonomie beaucoup plus simple.

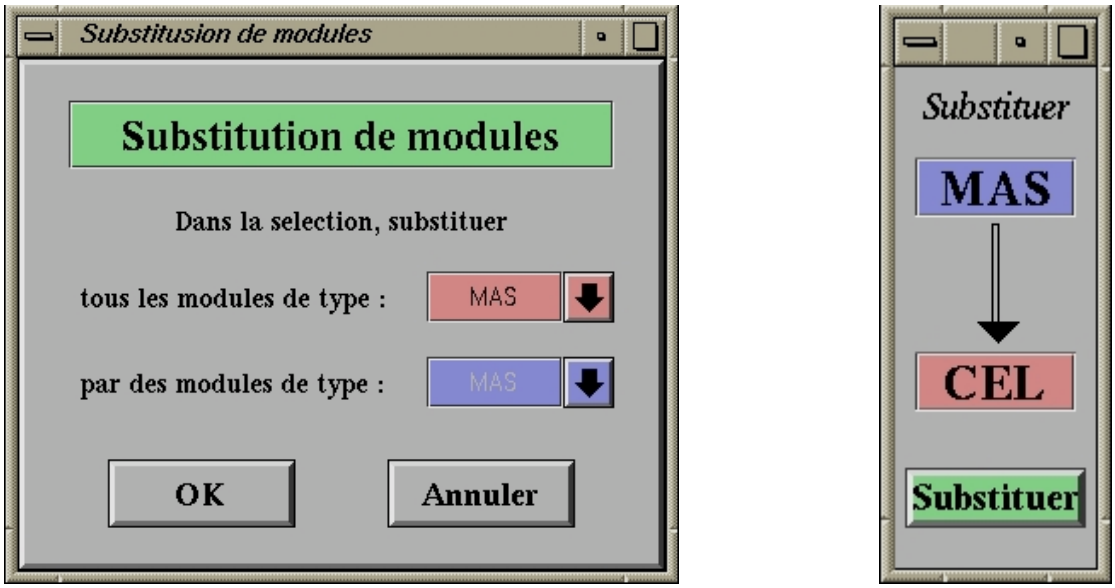


Figure 17 : fenêtre de substitution *(Couleur p. 446)***.**
A gauche : dans GENESIS 1.3. A droite : dans GENESIS 1.5.
L'évolution de la fenêtre montre combien une ergonomie gagne à être épurée.

18.3 - Multiplication des paramètres et maîtrise des échelles

En discutant du processus de création, nous avons insisté sur l'importance de la maîtrise des échelles et indiqué que l'échelle d'un objet peut être caractérisée sous trois angles :

- L'échelle d'amplitude ;
- L'échelle des constantes de temps (fréquence et temps d'amortissement).
- L'échelle d'impédance ;

L'échelle d'amplitude concerne, essentiellement, les conditions de l'excitation de l'objet considéré et donc les conditions initiales. Les échelles d'impédance et des constantes de temps dépendent, quant à elles, des valeurs des paramètres physiques.

Dans le chapitre précédent, nous avons vu qu'en multipliant les paramètres de raideurs (resp. de viscosité) d'un objet par un coefficient, il est possible de décaler le spectre de ses modes sur l'axe des fréquences (resp. de multiplier tous les temps d'amortissements). De même, multiplier l'ensemble des paramètres M, K et Z d'un objet par un *même* coefficient n'affecte en rien ses propriétés intrinsèques mais modifie sa capacité à collaborer avec son environnement, c'est à dire son échelle d'impédance[1].

[1] Si l'objet est composé de module non linéaires LNL, il faut pour modifier son impédance sans affecter ses propriétés intrinsèques multiplier également la force exercée par le LNL par le même coefficient.

De façon plus générale, il apparaît :

- Que maîtriser les échelles suppose de contrôler les rapports entre les paramètres M, K et Z à l'intérieur d'un même objet et entre les différents objets posés sur l'établi.
- Que le fait d'appliquer un coefficient multiplicateur à certains paramètres d'un objet peut permettre la modification de certaines de ses échelles tout en garantissant la conservation de certaines propriétés.

Puisqu'elle affecte précisément les *rapports* entre paramètres (d'un même objet ou entre objets), la *multiplication* est la principale opération algébrique sur les paramètres qui fasse sens. Une fonctionnalité lui est dédiée dans GENESIS.

Après plusieurs prototypes, nous avons retenu une ergonomie matricielle (figure 18). Dans la fenêtre de multiplication des paramètres, l'utilisateur est invité à choisir les paramètres physiques qu'il souhaite affecter dans un tableau à deux dimensions faisant apparaître les types de modules et leurs paramètres, à déterminer le coefficient multiplicateur puis à valider les choix effectués.

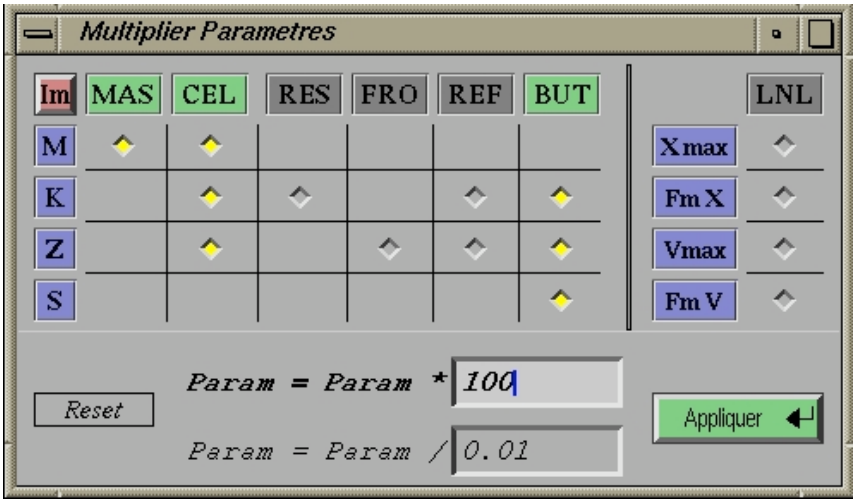


Figure 18 : la fenêtre de multiplication des paramètres de GENESIS 1.5.
(Couleur p. 446).

Le bouton **Im** (en haut à gauche du tableau) permet de sélectionner d'un clic tous les paramètres du tableau, assurant que *l'impédance* de la sélection sera modifiée sans que ses autres propriétés ne soient affectées.

Chapitre 19

PERSPECTIVE POUR LES FONCTIONNALITES EVOLUEES

Doté de ses fonctionnalités fondamentales et évoluées, l'atelier de lutherie de GENESIS 1.5 offre d'ores et déjà un très large champ d'investigation. Les fonctionnalités disponibles sont cohérentes et se complètent efficacement. Certaines cependant gagneraient à être développées plus avant, et d'autres devraient être ajoutées. Dans ce dernier chapitre, nous synthétisons l'ensemble des évolutions qui devraient permettre d'amener à moyen terme le mode lutherie de GENESIS à un degré de finalisation optimal. Chacune de ces évolutions a d'ores et déjà donné lieu à une réflexion approfondie et à un ou plusieurs cahiers des charges. Certaines d'entre elles existent déjà à l'état de prototype, mais n'ont pas donné entière satisfaction et appellent de plus amples réflexions. Nous aborderons successivement :

- La notion d'éditeur évolué de paramètres et le système de relations et de meta-paramètres.
- Les évolutions prévues pour les Ensembles et les Capsules qui soutiennent la démarche macro-modulaire de l'utilisateur ;
- Plusieurs fonctionnalités plus simples qui devraient, dans un proche avenir, être intégrées à GENESIS.

19.1 - Edition évoluée des paramètres ; relations et meta-paramètres

19.1.1 - Motivation

L'observation d'utilisateurs in-situ permet de se rendre compte que la maîtrise des paramètres dans GENESIS est à la fois un point clé du succès et une source de difficulté. L'expérience, comme la réflexion théorique, montrent par ailleurs que les paramètres des objets doivent pourvoir être envisagés globalement : la modification d'un paramètre pour obtenir un effet recherché (par exemple : régler la fréquence d'une cellule CEL) induit souvent des effets de bords qui vont modifier d'autres propriétés de l'objet (par exemple l'impédance, le temps d'amortissement, etc.). L'obtention d'un comportement satisfaisant demande que des combinaisons entre plusieurs paramètres soient considérées.

De la même manière qu'il n'y a guère d'intérêt à considérer un module isolément, la valeur d'un paramètre ne vaut souvent que lorsqu'elle est envisagée en *relation* avec d'autres. La notion de *relation* est ici un concept central. Tout utilisateur expert de GENESIS acquiert progressivement un savoir-faire quant aux relations entre valeurs de paramètres. Dans GENESIS 1.5, il n'existe cependant pas de fonctionnalités qui soutiennent ou permettent de développer un tel savoir-faire.

Notons que la notion de *relation* ici évoquée « transcende » celle de relation des paramètres à l'intérieur d'une entité définie (comme un macro-module, par exemple). Il s'agit de relations qui peuvent toucher des rapports entre objets, à un niveau complexe, plus que les objets eux-même.

Parmi les évolutions importantes que nous prévoyons pour GENESIS figure l'introduction d'un éditeur évolué de paramètres. Au vu de l'analyse des tâches proposée dans la partie V, les enjeux de cet éditeur sont divers. Il devrait permettre à l'utilisateur :

- *d'expliciter* son savoir faire en matière de paramètres – de la même manière que les Ensembles et les Capsules lui permettent d'expliciter sa démarche macro-modulaire ;
- De définir des *meta-paramètres* par la mise en relation de plusieurs paramètres.
- D'éditer simultanément de nombreux paramètres en relations les uns avec les autres par l'utilisation d'un nombre réduit de meta-paramètres plus puissants et ainsi d'alléger le processus d'édition des paramètres.
- D'opérer un changement de base, et ce faisant de privilégier certains parcours dans l'espace des paramètres lorsqu'il est exploré.

Une utilisation typique de l'éditeur évolué serait de permettre un contrôle plus direct de certaines propriétés physiques des objets. Dans l'hypothèse ou une certaine propriété, mettons une qualité sonore (timbre, fréquence, etc.) serait fonction d'un ensemble de paramètres liés par une loi quelconque, il deviendrait alors possible de représenter cette loi dans GENESIS et donc de contrôler la propriété directement en choisissant une valeur pour le meta-paramètre ainsi défini.

Incidemment, un enjeu annexe de l'éditeur évolué de paramètres serait ainsi de 'condenser' dans certains meta-paramètres une partie de l'expertise des acousticiens, notamment développée au sein du laboratoire, pour la mettre à disposition de musiciens.

Enfin, l'éditeur évolué de paramètres nous semble être une nécessité pour que la notion de 'surface paramétriques' des Capsules puisse être implémentée avec toute la puissance requise. Ce point sera discuté dans le chapitre suivant consacré aux évolutions des fonctionnalités macro-modulaires.

Un prototype d'éditeur évolué a permis à la fois de prouver l'intérêt de telles fonctionnalités, de mieux comprendre leurs enjeux et de mettre en évidence enfin certaines difficultés quant à leur conception[1]. En nous basant sur les enseignements acquis lors du développement et des tests de ce prototype, nous pouvons, dans ce paragraphe, préciser les principales caractéristiques de ce que devrait être l'éditeur évolué des paramètres dans les futures versions de GENESIS.

19.1.2 - Relation et meta-paramètres

L'édition évoluée des paramètres suppose la conception d'un système complet pour la gestion de *meta-paramètres* et de *relations*.

Dans ce système, un *meta-paramètre* définit une *relation* ou expression algébrique reliant plusieurs paramètres ou conditions initiales d'un ensemble de modules. Le nom du meta-paramètre est choisi par l'utilisateur. Le système doit être réentrant, c'est-à-dire qu'un meta-paramètre défini sur l'ensemble des modules concernés devrait pouvoir entrer dans la définition d'un nouveau meta-paramètre. Le système doit en conséquence s'appuyer sur un *espace de noms* constitué de tous les paramètres de l'objet et des meta-paramètres qui y sont définis. Dans la suite, le terme 'paramètre' désignera aussi bien les paramètres et conditions initiales des modules CORDIS-ANIMA que les meta-paramètres.

[1] Version'1.4BIS – éditeur évolué' de GENESIS - voir [Torvik.01].

La définition des expressions doit pouvoir faire intervenir indifféremment :

- Les fonctions mathématiques usuelles sur les réels ;
- Des fonctions statistiques (à minima somme, moyenne et écart-type) ;
- Des fonctions ad hoc pour chacun des outils d'analyse de GENESIS ; une fonction renvoyant le résultat de l'analyse modale des modules concernés (et dont le calcul reposerait sur l'exécution du programme AnModTop d'analyse modale) est notamment indispensable.

Une relation peut être résolue. *Résoudre* une relation c'est, ayant fixé la valeur de certains des paramètres qu'elle implique, déterminer les valeurs des autres paramètres de telle sorte que l'expression soit vérifiée. L'objectif est que l'utilisateur ait toute liberté quant aux paramètres qu'il choisit de fixer lorsque la relation est résolue.

Quant à l'expression algébrique qui est au cœur d'une relation il convient :

- De lui donner un *statut* précisant les conditions de sa résolution. On peut notamment envisager qu'une relation ne soit résolue que *sur ordre*. A l'inverse on peut envisager qu'elle soit *active* et que GENESIS cherche à la résoudre dès que l'un des paramètres qu'elle implique est modifié.
- De lui adjoindre une *stratégie de résolution* qui précise, pour chacun des paramètres comment les autres paramètres doivent être calculés lorsque le premier est modifié. Des *contraintes* doivent pouvoir être spécifiées pour guider la résolution (par exemple d'homogénéité ou de conservation des proportions entre certains paramètres). La notion de stratégie de résolution, incidemment, devrait pouvoir être étendue au cas où plusieurs paramètres sont spécifiés simultanément.
- De lui adjoindre des données quant aux *valeurs possibles* des différents paramètres impliqués dans la relation (valeurs extremums, ensembles de valeurs discrètes, etc.) ;

Un exemple simple de relation est celle qui pourrait lier les paramètres de raideur d'une corde de la façon suivante :

```
Meta-paramètre :  inertie_totale
Expression        inertie_totale = MOYENNE(MAS[i].M, CEL[i].M)
Plage de valeur : quelconque
Stratégie :
    - modification de inertie_totale =>
                  modification de toutes les inerties
    - modification d'une inertie =>
                  modification de toutes les autres inerties pour conserver l'homogénéité
                        et calcul de inertie_totale.
```

Ainsi définie, une telle relation assurerait que la corde soit homogène et que son inertie totale soit connue et modifiable.

Un autre exemple un peu plus complexe pourrait être défini sur un ensemble de modules par :

Meta-paramètre :	freq_fond
Expression	freq_fond = ANALYSE_MODALE(TOUS_MODULES)
Plage de valeur :	freq_fond ∈ [100 ; 500]
Stratégie :	
- Modification de freq_fond =>	
	modification de toutes les raideurs
- Modification d'un paramètre quelconque =>	
	calcul de freq_fond

Ici, la relation est à l'image de la fonctionnalité d'analyse modale telle qu'elle est implémentée dans GENESIS 1.5. En cas de modification du meta-paramètre freq_fond, la résolution de la relation consisterait à accorder l'objet. En cas de modification de l'un des paramètres physiques (inertie, raideur, etc.), la résolution de la relation consisterait à recalculer freq_fond en procédant à une nouvelle analyse.

19.1.3 - Analyse des difficultés

Il est certain que le système de relations et meta-paramètres tel que nous l'avons introduit est ambitieux et pose de sérieuses difficultés.

La première d'entre elle est qu'il se prête mal à la mise en œuvre d'une démarche itérative telle que celle adoptée pour une grande partie de la conception de GENESIS. Il ne peut en effet présenter un intérêt pour l'utilisateur que parvenu à une certaine maturité, et inversement deviendrait vite un obstacle majeur à la prise en main du logiciel s'il était incomplet ou mal interfacé. Mais plusieurs problèmes délicats doivent être résolus pour que cette nécessaire maturité soit obtenue. En conséquence, le système de relations et meta-paramètres (ou l'édition évoluée des paramètres) nous semble constituer l'une des perspectives de recherche majeures pour les prochaines années autour du projet GENESIS.

Nous relevons ici quelques-uns des principaux problèmes techniques et scientifiques qu'il pose.

Un langage pour définir les relations

Tout d'abord, le système de relations et meta-paramètres suppose qu'un langage soit défini. Il n'y a pas là, techniquement parlant, de problème fondamental. La difficulté tient en fait à la définition des primitives du langage, qui doivent être adaptées à la problématique de la modélisation physique avec CORDIS-ANIMA. Le prototype d'éditeur évolué qui a été conçu reposait exclusivement sur les opérateurs élémentaires (sommation, soustraction, multiplication et division) et les fonctions mathématiques usuelles (telles la racine carrée et les fonctions logarithmiques). Il conviendrait, à minima, d'ajouter à cette base des fonctions statistiques et une fonction représentant l'analyse modale que propose GENESIS. En outre, les paramètres doivent pouvoir être désignés par types (toutes les raideurs, toutes les viscosités, etc.), par module (raideur d'un module, etc.) ou par ensemble de modules (inerties des modules MAS, inerties des… etc.). Enfin, à ce langage permettant de définir les meta-paramètres par des expressions, il faudrait encore ajouter des fonctionnalités permettant de spécifier les contraintes que devraient vérifier les paramètres impliqués dans la relation.

Un moteur pour la résolution des relations

La résolution des relations pose un autre type de problème. Elle suppose d'abord que les expressions qui les définissent puissent être inversées. Cela, déjà, pose quelques problèmes, puisqu'il n'est pas évident que la fonction utilisée soit bijective et l'expression inversable de façon unique. Ce type de problème peut cependant être résolu, comme l'a montré le prototype d'éditeur évolué.

Mais l'inversion simple des relations n'est pas suffisante. La résolution des relations devrait également pouvoir faire intervenir des calculs d'optimisation qui, lorsqu'un nombre restreint de valeurs souhaitées par l'utilisateur est fixé, permettrait de calculer *l'ensemble* des valeurs des autres paramètres de sorte que la relation soit vérifiée ou au minimum approchée.

Enfin, de tels calculs d'optimisation devraient également pouvoir être étendus à la résolution simultanée de plusieurs relations, certains des paramètres qu'elles impliquent étant fixés.

Il n'est guère besoin de plus de détail pour que l'ampleur des difficultés adressées par la résolution des relations soit perceptible. Elle suppose, de fait, qu'un moteur de résolution complexe soit étudié et implémenté.

Des problèmes essentiels de nature ergonomique

L'un des enjeux majeurs de l'éditeur évolué est de faciliter l'édition des paramètres, qui au vu des remarques d'utilisateurs constitue une tâche particulièrement difficile. La complexité inhérente aux principes de l'éditeur évolué risque cependant, si l'on n'y prend garde, d'induire pour l'utilisateur de nouveaux types de difficultés tout aussi problématiques.

Tout d'abord, l'éditeur évolué fait interférer le principe de 'manipulation directe' qui, jusqu'à la version 1.5, prédomine largement, avec un style d'interaction plus abstrait. L'introduction de ce nouveau style d'interaction nécessite une certaine intelligence ergonomique afin que l'articulation entre les différentes *modalités* se fasse naturellement.

Par ailleurs, il faut rappeler ici que les utilisateurs de GENESIS, en effet, ne sont à priori ni mathématiciens ni, de façon plus générale, scientifiques. Il est certain que les expressions mathématiques qui constituent le cœur des relations ne peuvent, pour une grande partie d'entre eux, qu'être difficile à appréhender et plus encore à concevoir. L'éditeur évolué doit faire preuve d'une certaine 'transparence', difficile à obtenir compte tenu de sa complexité.

Il nous semble nécessaire qu'il offre plusieurs niveaux de prise en main. A minima, la *définition* des relations doit être clairement séparée de leur *utilisation*. Il devrait être possible à l'utilisateur de bénéficier de relations et meta-paramètres prédéfinis sans nécessairement qu'il doive en comprendre tout le fonctionnement.

Pour ce qui est de la définition des relations, il nous semble indispensable qu'un 'catalogue de relations' soit accessible, afin que les expressions les plus courantes n'aient plus qu'à être adaptées à l'ensemble des modules auxquels elles s'appliquent.

En résumé, nous considérons que l'éditeur évolué nécessite, en plus d'une recherche théorique et technique, une recherche ergonomique conséquente.

19.2 - Evolution des fonctionnalités de macro-modularité

Si les concepts d'Ensemble et de Capsule répondent en théorie aux multiples enjeux de la démarche macro-modulaire, il est certain que l'implémentation qui en est faite dans GENESIS 1.5 doit être complétée. Nous précisons dans ce paragraphes les différentes évolutions qui nous semblent nécessaires.

19.2.1 - Evolution des Ensembles

Nous distinguons, tout d'abord, quatre évolutions pour les Ensembles.

Représentation des Ensembles

Un développement envisageable dès les prochaines versions concerne la *représentation* des Ensembles sur l'établi. La principale difficulté est ici de déterminer une stratégie de représentation qui permette de faire apparaître sur l'établi les étiquettes des Ensembles sans alourdir la représentation des objets. On pourrait afficher en toutes lettres sur l'établi les noms des Ensembles, à la manière de légendes. Ceci fait, la représentation des Ensembles devrait être une aide précieuse pour guider la navigation dans les objets complexes et faciliter leur compréhension.

Organisation des étiquettes

L'organisation des étiquettes dans la fenêtre de gestion des Ensembles de GENESIS 1.5 se limite au choix de l'ordre dans lequel elles apparaissent. Cette possibilité est insuffisante. Dès lors qu'un nombre important d'Ensembles existe dans un objet, il devient en effet difficile de repérer dans la fenêtre un Ensemble particulier.

Deux pistes sont intéressantes en matière d'étiquetage des ensembles.

Tout d'abord, il serait utile de pouvoir n'afficher dans la fenêtre que certains Ensembles. Ceci pourrait se faire sur la base de critères de sélection (Ensembles dont le contenu est protégé, Ensembles dont le contenu a été analysé, etc.).

Ensuite, il semble nécessaire de pouvoir expliciter les liens fonctionnels existant entre certains Ensembles, par exemple : toutes les structures vibrantes, tous les ensembles d'un sous objet, tous les instrumentistes jouant sur une structure vibrante particulière, etc.

Il s'agit donc en quelque sorte de faire apparaître une 'arborescence'. Dans celle ci, cependant, un Ensemble donné doit pouvoir figurer en différents nœuds. Un déclencheur, par exemple, peut appartenir à la fois à l'ensemble des déclencheurs de l'objet et à l'ensemble des instrumentistes de la structure vibrante qu'il déclenche.

Une solution, relativement simple à mettre en œuvre, pourrait être d'autoriser qu'un Ensemble puisse figurer au contenu d'un autre Ensemble. Cette possibilité est, compte tenu de l'architecture adoptée, envisageable à court terme.

Protection du contenu et autres propriétés

La possibilité de protéger les Ensembles fait partie des caractéristiques que nous avons considérées dès leur définition. Elle n'est cependant pas encore mise en œuvre dans GENESIS.

La protection des Ensembles doit pouvoir concerner diverses propriétés. Un choix intéressant pourrait être de proposer :

- Une protection structurelle qui interdise la suppression de modules, la modification de connexions, ainsi que toute modification du contenu de l'ensemble (ajout ou suppression de modules de la liste).
- Une protection 'graphique' qui interdise que les positions relatives des différents MAT de l'Ensemble soient modifiées ;
- Une protection des paramètres qui interdise toute modification des paramètres des modules de l'Ensemble.

En ce qui concerne les paramètres d'autres types de protection pourraient être intéressantes. On pourrait souhaiter par exemple une protection des rapports entre certains paramètres, pour assurer la conservation de l'homogénéité des paramètres sur un Ensemble. Il est cependant très difficile ici de proposer une liste exhaustive de fonctionnalités. Ce type de protection évoluée, pour pouvoir être pratiquée dans toutes les situations, gagne à être envisagé non pas au niveau des ensembles mais au niveau du système de relations et meta-paramètres, bien plus flexible dès lors qu'il s'agit de travailler sur les paramètres.

19.2.2 - Evolutions des capsules ; surface paramétrique

A l'heure actuelle, lorsqu'une capsule est créée, la seule façon de modifier les paramètres des modules qui la constituent est de l'ouvrir dans un nouvel établi – donc d'accéder au contenu que la capsule a justement pour but de cacher. L'évolution majeure qui ferait accéder la Capsule au statut de macro-module serait le développement de la notion de surface paramétrique. Elle devrait reposer, d'une manière ou d'une autre, sur le système de relations et meta-paramètres.

19.3 - Autres fonctionnalités

Plusieurs autres fonctionnalités devraient à terme être introduites pour compléter les possibilités de GENESIS. Nous en évoquerons quatre parmi les plus importantes.

19.3.1 - Substitution évoluée

L'outil de substitution proposé par GENESIS 1.5 ne travaille qu'au niveau des modules élémentaires. Un outil de substitution évoluée est souhaitable pour permettre la substitution d'un ensemble de modules à une partie quelconque d'un objet. La substitution évoluée supposerait trois étapes :

- Préciser la structure du réseau MAT/LIA qui doit faire l'objet de la substitution ;
- Sélectionner les modules qui devront être substitués ; dans la sélection, la structure précédente peut apparaître plusieurs fois.
- Choisir l'ensemble de modules appelé à remplacer la structure en question à chaque fois qu'elle apparaît dans la sélection ;
- Spécifier quels points M et L de cet ensemble de modules devront se substituer aux points M et L à la 'surface' de la structure faisant l'objet de la substitution ;
- Procéder à la substitution.

19.3.2 - Génération évoluée

Un travail théorique conséquent concerne l'extension du pont que doit proposer GENESIS entre l'univers des objets CORDIS-ANIMA d'une part et celui de leurs propriétés d'autre part.

Ce travail s'inscrit dans la lignée de l'analyse modale, mais concerne plus spécifiquement les outils de *génération* évoluée. Il s'agit en fait du " problème inverse".

L'objectif est d'aboutir à la définition de fonctionnalités qui, à partir d'un ensemble souhaité de propriétés (contenu modal, impédance, etc.) et de contraintes (taille de l'objet, structure souhaitée pour le réseau de modules, contraintes quant à l'homogénéité des paramètres, etc.) permettraient de générer un objet CORDIS-ANIMA qui les produise.

Plusieurs prototypes indépendants de GENESIS ont été conçus successivement par différents chercheurs et étudiants, sans qu'aucun ne soit à ce jour satisfaisant. Il faut dire que le problème est complexe et nécessite une analyse mathématique poussée des propriétés des réseaux CORDIS-ANIMA. Une synthèse de ces différents prototypes, ainsi qu'un travail pour les intégrer à GENESIS, devrait permettre dans un avenir proche de mettre à la disposition de l'utilisateur une première version de l'outil de génération évoluée.

19.3.3 - Exploration paramétrique systématique

La partie V a montré que l'exploration systématique de l'espace des paramètres d'un objet tient une place essentielle dans le processus de création avec CORDIS-ANIMA. Pour mener une telle exploration dans GENESIS 1.5, l'utilisateur doit spécifier 'à la main' chacun des modèles qu'il souhaite explorer, puis procéder aux simulations de chacun de ces modèles. Cela s'avère rapidement fastidieux et limite considérablement la portée des explorations paramétriques effectivement réalisées. Il convient d'automatiser le processus.

RESUME ET CONCLUSION

Dans cette VIe partie, nous avons présenté les fonctionnalités du mode lutherie de GENESIS 1.5 qui permettent un travail évolué sur les objets :

- Les concepts d'Ensemble et de Capsule venant soutenir la démarche macro-modulaire de l'utilisateur ;
- L'analyse modale et l'accordage qui constituent les premiers éléments d'un ensemble d'outils orientés vers un contrôle explicite des propriétés des objets.
- Plusieurs fonctionnalités complémentaires d'édition, simples mais cependant indispensables à l'utilisabilité du logiciel.

Nous avons enfin évoqué les principales perspectives de développement à court ou moyen terme, complétant ainsi la vue d'ensemble de ce que serait, selon nous, une version optimale de l'atelier de lutherie GENESIS.

Partie VII

Simulation et phénomènes

Jusqu'ici, nous avons traité des fonctionnalités soutenant la phase de *Lutherie*.

Les fonctionnalités de simulation présentées dans cette dernière partie ont par nature un statut particulier dans le processus de création et, par voie de conséquence, dans GENESIS. Qui plus est, leur implémentation et leur ergonomie sont particulièrement sensibles.

Le premier chapitre est consacré à la *fenêtre de simulation* de GENESIS 1.5. Il présente ses fonctionnalités et son ergonomie ainsi que les raisons des choix effectués pour leur mise en œuvre.

Dans le second chapitre, nous examinons les perspectives de développement des fonctionnalités de simulation pour les versions futures de GENESIS.

Introduction

Contexte et bibliographie

Partie I : le paradigme de la simulation interactive et multisensorielle d'objets physiques

Partie II : Interaction Homme-Machine

Partie III : Le processus de création avec GENESIS. Analyse des Tâches

Conception de GENESIS

Partie IV : Choix généraux pour l'environnement GENESIS

Partie V : Fonctionnalités fondamentales de l'atelier de lutherie GENESIS

Partie VI : Fonctionnalités évoluées du mode lutherie

Partie VII : Simulation et Phénomènes

Chapitre 20 La fenêtre de simulation 322

1.1 - LA SIMULATION DANS GENESIS 1.2 A 1.4 322
1.2 - ANALYSE DES ENJEUX ET DES TACHES 324
1.3 - FONCTIONNALITES ET ERGONOMIE DE LA FENETRE DE SIMULATION 327
1.4 - NOTES SUR L'IMPLEMENTATION 335

Chapitre 21 Perspectives pour les fonctionnalités de simulation 338

2.1 - FINALISATION 338
2.2 - METROLOGIE 340
2.3 - VERS UNE VERITABLE MULTISENSORIALITE ; CONNEXION AU MOTEUR TEMPS-REEL TELLURIS 341

Conclusions et perspectives

Chapitre 20

LA FENETRE DE SIMULATION

Les fonctionnalités de simulation sont particulièrement sensibles tant en termes techniques que parce qu'elles conditionnent le potentiel de *l'ensemble* de l'environnement. Elles ont donc donné lieu à une démarche itérative poussée impliquant plusieurs prototypes, versions et campagnes de tests.

Le premier paragraphe présente les principales conclusions de ce cheminement.

Dans les paragraphes suivants nous résumons les enjeux et problèmes relatifs aux fonctionnalités de simulation à travers une *analyse* exhaustive *des tâches*. Nous présentons ensuite la *fenêtre de simulation* de GENESIS 1.5, ses fonctionnalités et son ergonomie. Enfin nous donnons quelques précisions quant aux principes et procédés techniques mis en œuvre.

20.1 - La simulation dans GENESIS 1.2 à 1.4

Dès la version 1.2, deux types de simulations cohabitent dans GENESIS. *Ecouter* un objet, c'est lancer une simulation dans le but exclusif de générer un signal sonore. *Regarder* un objet, c'est afficher pas à pas son *état* dans une fenêtre graphique.

Les simulations *Ecouter* sont effectuées le plus rapidement possible. Le signal sonore est normé à la fin du calcul. Il peut alors être lu et le cas échéant visualisé.

Lors d'une simulation *Regarder*, l'état de l'objet est affiché au fur et à mesure de son calcul. La vitesse du calcul s'en trouve considérablement ralentie. Ainsi, des phénomènes dont les fréquences caractéristiques sont dans le domaine audible deviennent perceptibles et analysables à l'œil.

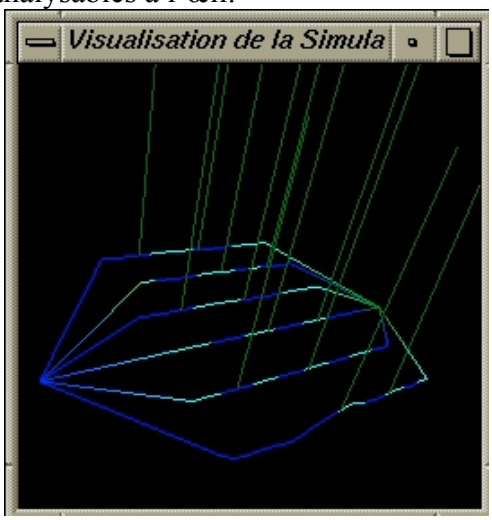


Figure 1 : un objet en mouvement (GENESIS 1.2)

Comme nous l'indiquions dans la partie V, la représentation d'un objet en mouvement ne peut être faite dans le plan de l'établi. Aux deux dimensions de l'espace topologique il faut ajouter l'axe perpendiculaire des mouvements. C'est l'unique axe suivant lequel les <MAT> de l'objet se déplacent. L'objet simulé est ainsi représenté dans un espace tridimensionnel (figure 1).

Les <MAT> sont visualisés par des sphères (optionnel). Les <LIA> sont eux figurés par des traits reliant les positions des <MAT> qu'ils connectent.

Ce dernier choix tend à faire croire que les forces ont trois composantes, alors qu'elles ne s'exercent que suivant l'axe des mouvements (figure 2). Il permet en revanche un graphisme simple et efficace et a été conservé dans la version 1.5.

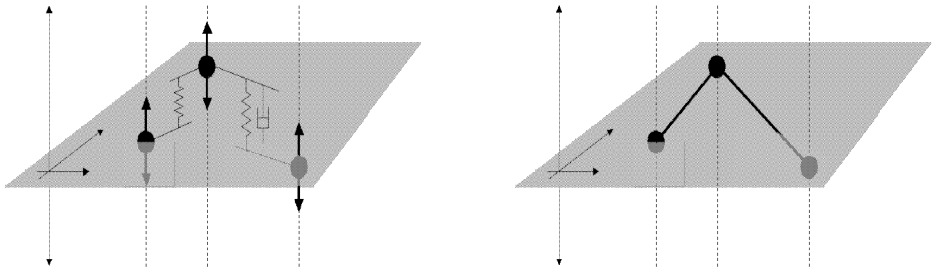


Figure 2 : deux représentations possibles des <LIA>.
Celle de gauche est plus proche de ce que réalise effectivement la simulation, mais alourdit considérablement le graphisme et nuit à sa lisibilité.
Celle de droite est celle adoptée lors des simulations Regarder.

De façon plus générale, il est apparu qu'un utilisateur novice éprouve des difficultés à conceptualiser l'espace tridimensionnel et la façon dont il est organisé. L'ajout de l'axe des mouvements au plan de l'établi pose problème. De plus, dans les versions 1.2 à 1.4, il est difficile de parvenir à un point de vue satisfaisant sur l'objet.

Il nous est donc apparu *essentiel* de concevoir dans la version 1.5 une ergonomie qui facilite autant que faire se peut la compréhension de l'espace tridimensionnel et qui rende la navigation plus intuitive. Malgré les efforts produits en ce sens et les résultats pertinents obtenus, nous considérons que la représentation des objets dans l'espace 3D nécessite, encore aujourd'hui, d'être introduite avec attention lors de séances de formation ou dans les supports pédagogiques (mais peut-il en être autrement ?). Elle demeure, avec le formalisme topologique, l'un des « points sensibles » de GENESIS, c'est-à-dire qu'elle est susceptible d'induire, lorsqu'elle est mal abordée, un modèle mental « superstitieux » (au sens de Norman, [Norman.90]). Toutefois, ce cap franchi, elle devient satisfaisante.

En résumé, les versions 1.2 à 1.4 ont défini plusieurs principes de base concernant la simulation dans GENESIS. Ces principes nous apparaissant *nécessaires*, ils ont été conservés dans la version 1.5. Il s'agit de :

- La cohabitation de deux fonctionnalités distinctes : Ecouter et Regarder ;
- La succession (Simulation / Conversion / Ecoute) pour la fonctionnalité Ecouter ;
- Le principe de l'affichage *en flux* (pendant le calcul même) pour les simulations Regarder.
- Le principe, enfin, d'un axe des mouvements perpendiculaire au plan de l'établi et donc d'un espace tridimensionnel pour la représentation des objets en mouvement.

En revanche, les fonctionnalités de simulation dans les versions 1.2 à 1.4 sont loin d'être *complètes*. Qui plus est, elles ne sont pas non plus ergonomiques : ni *cohérentes* avec le reste de l'environnement, ni *affordantes*, ni intuitives[1]. Tout recours intensif à la simulation est dès lors rendu difficile. Sur la base de ces critiques et conformément à nos prévisions, nous avons pu entreprendre une refonte complète des fonctionnalités de simulation dans la version 1.5 de GENESIS. La conception de ces fonctionnalités s'est appuyée sur une analyse exhaustive des tâches, enrichie de l'observation répétée d'utilisateurs *in situ*. Le paragraphe suivant propose une synthèse.

[1] Dans la version 1.4 de nombreuses fonctionnalités ont été ajoutées et une *mise à plat* de l'ergonomie a été effectuée (chapitre 5). Ainsi, cette version a *rendu possible* le travail, même s'il ne s'effectue pas dans des conditions optimales. Plus, elle a permis d'observer attentivement des utilisateurs terminaux *in situ*, nourrissant ainsi notre réflexion pour la refonte des simulations dans GENESIS 1.5.

20.2 - Analyse des enjeux et des tâches

20.2.1 - Trois enjeux, trois modalités d'observation

L'analyse du processus de création a fait apparaître trois enjeux pour la simulation dans GENESIS. Il s'agit de :

- *Simuler pour comprendre* le comportement de l'objet conçu puis le cas échéant, fort de cette compréhension, pour affiner cet objet.
- *Simuler pour évaluer* l'intérêt de l'objet conçu. La simulation est alors au cœur de la *démarche itérative* de l'utilisateur, rappelée sur la figure ci-contre. Ici, il est important de pouvoir comparer les comportements de plusieurs objets afin d'élire le (ou les) plus intéressant.

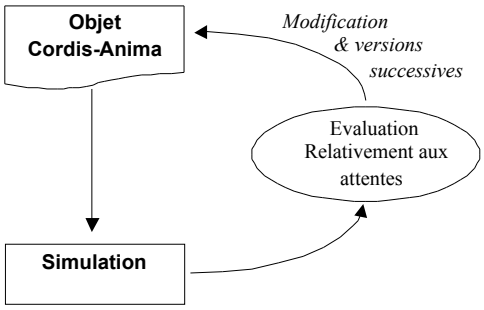


Figure 3 - rappel : la boucle modification / simulation / évaluation

- *Simuler pour terminer* un travail. La conception de l'objet en phase de lutherie vise, à terme, l'obtention d'un son à caractère musical. Elle se termine donc par une simulation et le cas échéant par l'exportation du son obtenu.

Par ailleurs trois attitudes sont possibles autour de la simulation d'un objet CORDIS-ANIMA. Chacune correspond à une *modalité d'observation* des phénomènes générés. On peut d'abord s'intéresser au signal sonore (c'est la fonctionnalité Ecouter). On peut ensuite observer l'objet en cours de mouvement (Regarder). On peut enfin vouloir procéder à des mesures sur les phénomènes (Mesurer). Le tableau de la figure 4 montre comment les trois enjeux (Evaluer, Comprendre, Terminer) se répartissent sur les trois modalités d'observation.

	Comprendre / affiner	**Evaluer**	**Terminer**
Ecouter	+	++	++
Regarder	++	-	-
Mesurer	++	+	-

Figure 4 : répartition des enjeux suivant les modalités d'observation des simulations

Chacune des modalités d'observation des phénomènes suppose un ensemble de tâches spécifiques, et par voie de conséquence nécessite un ensemble d'outils et une ergonomie qui lui est propre.

Dans GENESIS 1.5 seules quelques fonctionnalités très limitées sont proposées pour la mesure des phénomènes ; elles sont appelées à être développées ultérieurement, conformément à notre démarche incrémentale et seront donc examinées dans le chapitre II. Le reste de ce chapitre est consacré exclusivement aux modalités Ecouter et Regarder.

Les deux paragraphes qui suivent résument pour chacune des modalités Ecouter et Regarder, les tâches dont GENESIS devrait permettre l'accomplissement. L'ensemble de ces tâches a été pris en considération pour la re-conception du *noyau fonctionnel*, de l'architecture et de l'ergonomie des simulations de la version 1.5.

20.2.2 - Tâche pour la fonctionnalité Ecouter

Tâches générales

- Lancer une simulation. Stopper une simulation.
- Lancer plusieurs simulations sur un ensemble d'objets, de telle sorte qu'elles soient effectuées en parallèle ou successivement.
- Simuler un objet complet ou au contraire une partie seulement de cet objet.
- Savoir à quel objet ou partie d'objet se rapporte le signal sonore qu'on écoute ou qu'on visualise. Pouvoir disposer de cet objet sur un établi pour l'étudier, le modifier, etc. Associer un objet à un son ou un son à un objet.
- A l'inverse, exporter un son pour qu'il acquière une « vie autonome ».
- Naviguer parmi les sons préalablement synthétisés et figurant sur le compte utilisateur (retrouver un objet en partant des sons ou un son en partant d'un objet).

Durée des sons synthétisés

- Spécifier la durée du son à synthétiser en seconde ou en nombre d'échantillons.
- Augmenter pendant le calcul ou après le calcul, la durée du son à synthétiser.
- Relancer la simulation d'un objet à partir de l'instant ou une simulation précédente s'était arrêtée, dans la même session de travail ou lors de sessions successives.
- Diminuer la durée du son synthétisé pendant le calcul ou après le calcul. Diminuer la durée d'un fichier son[1].
- Estimer le temps de calcul restant avant l'obtention de la durée prévue.

Ecoute des sons

- Normer le son au maximum d'amplitude ou à une amplitude donnée[2].
- Ecouter le son depuis le premier échantillon ou depuis un échantillon particulier.
- Ecouter la partie du fichier son déjà calculée alors que la simulation se poursuit.
- Ecouter toutes les voies ou seulement certaines voies du fichier son.
- Stopper l'écoute d'un son.

Visualisation du signal sonore

Ici figurent une partie des tâches permises par les éditeurs de son courant quant à la visualisation du signal sonore. Précisons toutefois que dans le contexte de GENESIS tout travail ou modification directe du signal est exclu.

- Visualiser le signal sonore synthétisé.
- Visualiser toutes les voies ou seulement certaines voies du fichier son.
- Naviguer dans la représentation : zoom avant, zoom arrière, position, etc.
- Connaître l'amplitude de chaque échantillon, i.e. la position du module <MAT> qui a été « écouté » (ou de la force qui a été enregistrée)[3].
- Afficher sur cette représentation les instants importants : durée effective du signal sonore, durée prévue, barre de défilement pendant l'écoute, etc.

---

[1] Certains objets qui font intervenir un nombre important de non-linéarités sont susceptibles d'être globalement stables puis, subitement, de diverger. Il est alors essentiel de pouvoir restreindre la durée d'un son calculé à la partie stable de la simulation, sans avoir pour ce faire à relancer une simulation complète.

[2] Les échantillons des fichiers son générés par le simulateur sont codés en virgule flottante. Leur amplitude moyenne dépend notamment des conditions d'excitation de l'objet et peut être très variable suivant les conventions adoptées par l'utilisateur. L'écoute du signal sonore suppose donc qu'il soit préalablement normé. La normalisation au maximum d'amplitude permet un rapport Signal/Bruit minimal. La normalisation à un coefficient donné permet, elle, de comparer à l'oreille les amplitudes relatives des sons de plusieurs objets ou parties d'un même objet, dans une perspective de *composition* de ces objets.

[3] C'est l'une des fonctionnalités très élémentaires proposées par GENESIS 1.5 pour la *mesure* des phénomènes. Elle est utile pour comparer les amplitudes des mouvements de plusieurs objets.

20.2.3 - Tâches pour la fonctionnalité Regarder

Tâches générales

- Lancer une simulation. Stopper une simulation.
- Simuler plusieurs objet en parallèle. Choisir celui qui est visualisé[1].
- Simuler un objet complet ou une partie seulement de cet objet.
- Savoir à quel objet, quelle partie d'objet ou quel établi se rapporte la simulation visualisée. Pouvoir disposer de cet objet sur un établi pour l'étudier, le modifier, etc.
- Photographier l'objet en mouvement i.e. : créer un fichier image.

Contrôle du simulateur

- Afficher successivement chacun des pas de calcul, pour être certain d'observer tous les détails[2].
- Accélérer la simulation pour adapter l'affichage à la fréquence des phénomènes que l'on souhaite observer ou pour accéder plus rapidement à un instant donné de l'histoire de l'objet (avance rapide)[3].
- Ralentir la simulation pour mieux distinguer certains détails du mouvement de l'objet. Calculer et afficher pas à pas la simulation.
- Connaître la fréquence d'affichage par rapport au temps-réel.
- Choisir la fréquence d'affichage de telle sorte que les évolutions visualisées soit à une fréquence donnée par rapport au temps-réel (suivant ce que permet la complexité de l'objet : 1:1 (objet de petite taille), 1:10, 1:100, 1:1000 du temps réel, etc.).
- Accéder rapidement à un instant donné après lequel on souhaite observer ou ré-observer un phénomène particulier. Expliciter dans GENESIS les instants intéressants pour pouvoir y accéder plus tard rapidement.

Navigation

- Choisir le point de vue. Il s'agit ici de naviguer dans un espace tridimensionnel, c'est à dire *in fine* de choisir la position, la direction et l'angle d'ouverture d'une *caméra*.
- Adapter la plage d'amplitude visualisée à l'amplitude des mouvements de l'objet, i.e. : dilater ou comprimer l'axe X des mouvements indépendamment des deux autres axes de l'espace (ceux qui correspondent à l'établi).
- Visualiser une *partie* de l'objet. Régler le point de vue de telle sorte que cette partie soit visualisée de façon optimale. Demander que le point de vue soit calculé en ce sens.
- Accéder à des angles de vue standards (face établi, face axe des mouvements, zoom nominal, etc.). Initialiser la vue (de différentes manières).
- Enregistrer des vues préférentielles pour un objet ou pour l'environnement de travail i.e. pour chaque objet. Accéder aux vues préférentielles enregistrées. Echanger des vues préférentielles entre les objets.

---

[1] Nous n'avons jamais observé d'utilisateurs visualisant simultanément plusieurs objets, alors même que cela était possible avec la version 1.4 de GENESIS. La visualisation simultanée de plusieurs objets en mouvement ne fait pas partie des tâches de l'utilisateur. Elle n'est plus possible dans GENESIS 1.5.

[2] L'observation visuelle d'une simulation suppose qu'elle soit *ralentie* pour que les phénomènes de nature acoustique puissent être perceptibles et analysables par l'œil. Pour cela, le plus simple est d'afficher l'état de l'objet en *flux*, c'est-à-dire au fur et à mesure de son calcul effectué en temps-différé par le moteur CORDIS-OFF. Cette option s'est avérée satisfaisante dans les versions antérieures de GENESIS.

[3] Nous excluons l'option qui consisterait à enregistrer un *film* de la simulation (séquence des états successifs de l'objet), option trop coûteuse et à l'intérêt douteux. En conséquence de ce choix technique, « l'inversion du temps » (retour arrière) ne figure pas au nombre des tâches de l'utilisateur ici exposées.

Choix des options de visualisation

- Visualiser tout l'objet ou, au contraire, uniquement la partie qu'on souhaite observer pour clarifier l'écran – même si le reste est simulé.
- Visualiser les <MAT> et les <LIA> ou uniquement les <LIA>[1].
- Visualiser la grille de l'établi (ou non), les axes (ou non), etc.
- Choisir d'autres options de visualisation : lumière, texture. Visualiser en plein écran[2].

20.2.4 - Un quatrième enjeu pour l'ergonomie

Il nous faut enfin considérer un dernier point important pour la conception de l'ergonomie. Comme nous l'avons indiqué, les utilisateurs novices éprouvent certaines difficultés à conceptualiser l'immersion des objets topologiques dans l'espace tridimensionnel. La compréhension de l'espace 3D et de son articulation avec le plan topologique de l'établi est ardue. Il nous appartient donc de la soutenir, si possible.

20.3 - Fonctionnalités et ergonomie de la fenêtre de simulation

20.3.1 - La fenêtre de simulation

Dans GENESIS 1.5, les fonctionnalités Ecouter et Regarder sont regroupées dans une unique fenêtre de simulation, partagée par l'ensemble des établis ouverts. Ce choix appelle une rapide discussion.

Pour l'utilisateur, l'objet conçu s'identifie en première approche à la représentation qui en est proposée sur l'établi. Dans GENESIS 1.4, cependant, l'objet en mouvement évolue dans un espace différent et donne lieu à une représentation supplémentaire. La cohabitation de deux représentations semble s'opposer à ce que l'utilisateur puisse considérer qu'il voit et manipule un objet, par nature unique. Il parait préférable – et il serait possible – d'unifier ces deux représentations. Chaque établi pourrait alors *basculer* pour révéler l'axe de simulation (figure 5) et ainsi adopter deux états suivant que l'objet est édité ou Regardé.[3].

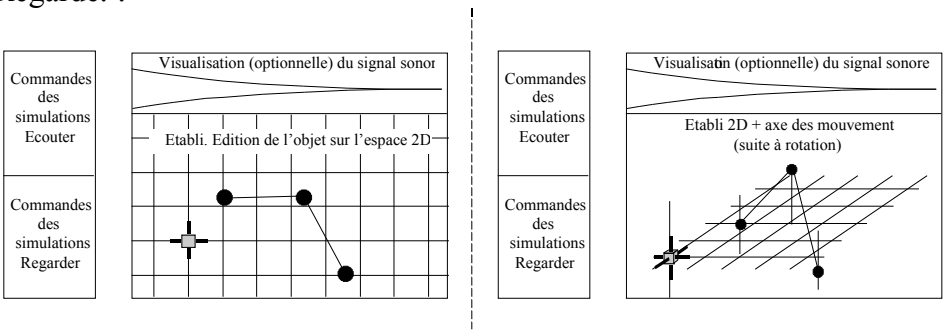


Figure 5 : une première option : visualtion de l'objet en mouvement dans l'établi

[1] Etant entendu que l'affichage des modules de type <MAT> est coûteux et tend à considérablement ralentir la simulation.

[2] La raison d'être essentielle de la visualisation reste de *comprendre* les phénomènes en jeu. Visualiser l'objet en mouvement peut cependant provoquer un certain sentiment esthétique qui peut être encore accentué par des jeux de lumière.

[3] Notons qu'un autre avantage plus prosaïque à l'unification des représentations est qu'elle réduirait le nombre *d'objets d'interface* présents à l'écran.

Cette option semble à première vue logique et naturelle. Elle n'a pourtant pas été retenue. Elle présente en effet des inconvénients majeurs, notamment de nature ergonomique. Parmi les principaux :

- Elle mettrait sur le même plan (en tout cas dans le même espace) l'édition de l'objet et l'observation des phénomènes qu'il génère. Elle rendrait la manipulation de l'établi – notamment la navigation – plus complexe.
- Elle tendrait à accentuer l'importance de la visualisation dynamique, certes très utile mais aussi rarement utilisée. A l'inverse, elle minimiserait la représentation plane de l'établi, alors même que nous voulons qu'elle soit prépondérante dans les images mentales de l'utilisateur.
- Elle interdirait l'édition de l'objet pendant qu'il est simulé. Elle interdirait de même que la représentation plane, aisément manipulable, puisse être utilisée pour agir « indirectement » sur la représentation 3D – piste qui nous le verrons est pourtant fort intéressante.
- Elle imposerait qu'à l'avenir deux ergonomies très différentes cohabitent suivant que la simulation est *temps différé* ou *temps réel*[1]. De façon similaire, elle nécessiterait une ergonomie radicalement différente pour les fonctionnalités de simulation au sein des futurs modes Gestion, Jeu et Composition, dans lesquels l'établi n'a plus lieu d'être.
- Enfin, la visualisation du signal sonore dans la même fenêtre que l'objet (figure 5 ci dessus) contredirait la métaphore de l'établi.

Compte tenu de l'ensemble de ces remarques, nous avons décidé de conserver deux espaces de représentation bien distincts.

Par ailleurs, nous avons également décidé de regrouper dans la même fenêtre les deux modalités d'observation Ecouter et Regarder. L'unicité de la fenêtre de simulation garantit que toutes les données phénoménologiques sont accessibles au même endroit. Elle offre de plus la possibilité d'une interaction entre la représentation du signal et celle de l'objet en mouvement.

La *fenêtre de simulation* est modulaire (figure 6 page suivante). Elle peut occuper une place réduite (en haut à gauche de la figure) ou prendre différents aspects suivant la ou les modalité(s) d'observation que veut privilégier l'utilisateur. A chaque espace correspond un ensemble de fonctionnalités que nous présentons dans les paragraphes suivants.

Pour finir notons que la fenêtre de simulation satisfait au principe de cohérence. C'est une fenêtre *libre* (non modale) qui peut rester affichée constamment, au même titre que la plupart des autres outils de GENESIS. Pour afficher les phénomènes relatifs à un objet donné, il faut et suffit de sélectionner l'établi ou il figure.

[1] Ceci pour des raisons techniques. Le moteur TELLURIS repose sur une architecture client/serveur, le serveur tournant sur une machine distante. Compte tenu des contraintes inhérentes à la communication inter-machines, il n'est pas envisageable que le client (ici : GENESIS) affiche l'état de l'objet calculé en temps réel par le serveur (TELLURIS). Il est donc impossible que la visualisation des simulations temps réel soit faite sur l'établi.

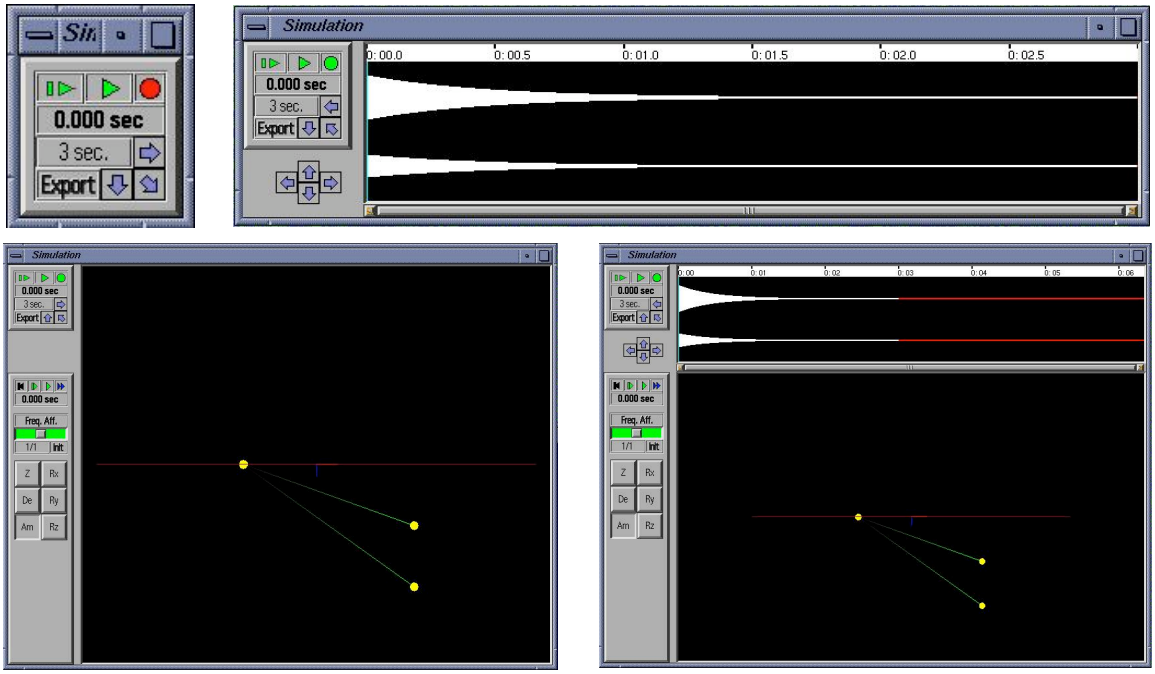


Figure 6 : la fenêtre de simulation est modulaire *(Couleur p. 447)**.***
Haut gauche : aspect élémentaire/timbre : l'écoute du son est privilégiée.
On peut repérer, au niveau du « timbre », les flèches de développement et de réduction.
Haut droit : représentation du signal
Bas gauche : observation de l'objet en mouvement (et écoute)
Bas droit : développement complet (objet en mouvement + signal).

Ainsi, les fonctionnalités simulations apparaissent au *même niveau* que les autres fonctionnalités. L'objet (via sa représentation sur l'établi) reste *central* et la simulation apparaît simplement comme un soutien à la tâche de l'utilisateur en phase de lutherie. L'objet est *simulé* de la même manière qu'il est *analysé* ou *ausculté* à l'aide d'une autre fonctionnalité. D'une certaine manière, les phénomènes deviennent une *propriété* de l'objet central, comme le sont les paramètres, les Ensembles ou les données modales.

20.3.2 - Conservation du lien causal

Dans les versions 1.2 à 1.4 de GENESIS, tout phénomène est coupé de sa cause. S'il veut conserver l'association entre objets et phénomènes, l'utilisateur ne peut s'appuyer que sur sa mémoire et sur des stratégies d'organisation de son espace de travail et des fichiers sur le compte utilisateur.

Pour apporter solution à ce type de problème, mais aussi plus fondamentalement pour favoriser si possible une *pensée physique*, nous avons là encore cherché à *placer systématiquement l'objet CORDIS-ANIMA* (ou du moins sa représentation) *au centre de l'interface et des tâches de l'utilisateur.*

Ainsi, dans GENESIS 1.5, les phénomènes n'ont d'existence qu'au travers des objets qui en sont la cause. Le lien de causalité est conservé jusqu'à ce que l'utilisateur demande explicitement que le phénomène (i.e. le signal sonore) soit exporté[1]. Il est archivé sur le compte utilisateur dans les fichier *.top* d'enregistrement des objets[2]. Pendant une session de travail, les phénomènes sont liés à l'objet qui les a générés : pour les observer, il faut (et

[1] L'exportation passe par une fenêtre modale. Rappelons qu'un son exporté est codé sous la forme d'entiers 16 ou 24 bits, 44100 Hz, au format AIFF (partie IV chapitre 2).

[2] Sous la forme d'un lien vers les fichiers secondaires – sons .echf ou .aiff, état .stt, etc.

suffit) de sélectionner l'établi correspondant. Ainsi, d'une certaine manière, on n'écoute plus un son (les sons, sauf cas exceptionnel, ne sont plus accessibles directement sur le compte utilisateur), mais on *écoute un objet* dont le signal sonore est une manifestation. De même, au processus qui consistait à naviguer parmi les sons d'un compte utilisateur est substituée celui qui consiste à naviguer parmi les objets et à écouter les sons résultant de leurs simulations– d'autres exemples pourraient être donnés.

20.3.3 - Ecouter

La figure 7 fait état de l'apparence du module Ecouter (+signal) de la fenêtre de simulation et de la répartition des diverses fonctionnalités correspondantes.

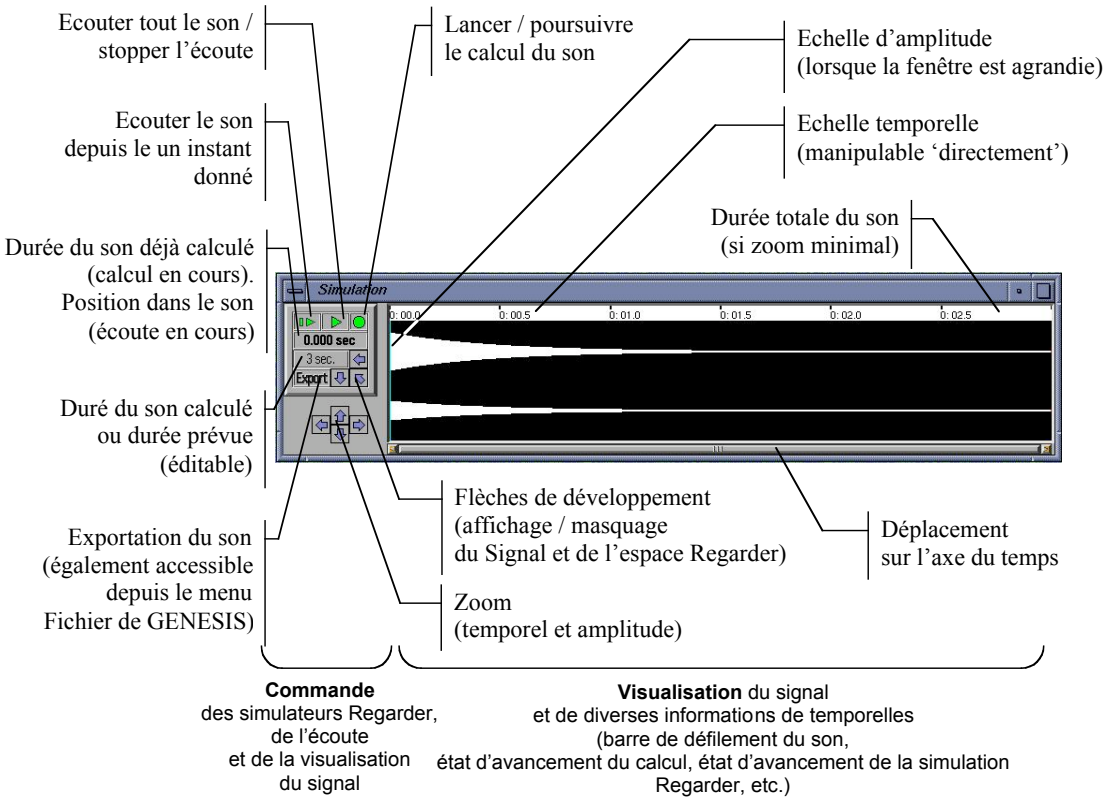


Figure 7 : la fonctionnalité Ecouter dans la fenêtre de simulation (GENESIS 1.5)

Ces fonctionnalités reposent essentiellement sur une interaction par *widget*, et dans une moindre mesure sur une manipulation directe de l'espace de représentation du signal. Certaines options (dont le type de normalisation à appliquer au son lors de l'écoute : maximum d'amplitude ou relativement à un coefficient donné) sont accessibles dans les préférences utilisateurs. Citons, enfin, quelques options complémentaires qui n'apparaissent pas dans la fenêtre :

- L'écoute (i.e. la lecture) du son d'un objet peut être déclenchée depuis l'établi, en actionnant la barre d'espace du clavier ; ce raccourci renforce l'idée que c'est l'objet qui est écouté, et non pas un signal hors de son contexte.
- L'écoute des sons est possible depuis la fenêtre d'ouverture des objets (menu fichier) ; incidemment, cela devra être étendu à l'ensemble des fonctionnalités d'organisation du compte utilisateur, qui constitueront à terme le mode Gestion.

Organisation de l'espace de représentation

Les choix opérés quant à l'espace de représentation de l'objet en mouvement visent essentiellement à faciliter la compréhension par l'utilisateur de la façon dont l'espace plan de l'établi et l'espace tridimensionnel s'articulent relativement l'un à l'autre.

Deux options étaient possibles pour l'orientation de l'axe de simulation (figure 8).

Dans la première, l'axe de simulation « sort » de l'écran. Pour le révéler en partant de la représentation plane de l'établi, il faut que le haut de l'établi bascule vers l'arrière. A cette première option correspond une *métaphore du tableau* des salles de classe' : l'utilisateur dispose d'une part d'une vue de face dans l'établi (le tableau), d'autre part dans la fenêtre de simulation d'une vue basculée à la manière de celle que l'élève a sur sa feuille.

Dans la seconde option, l'axe de simulation « rentre » dans l'écran. Dès lors, en « avançant » dans la profondeur de l'établi, l'utilisateur parcourt les X croissants de l'axe de simulation : nous dirons qu'à cette option correspond une *métaphore du marcheur*.

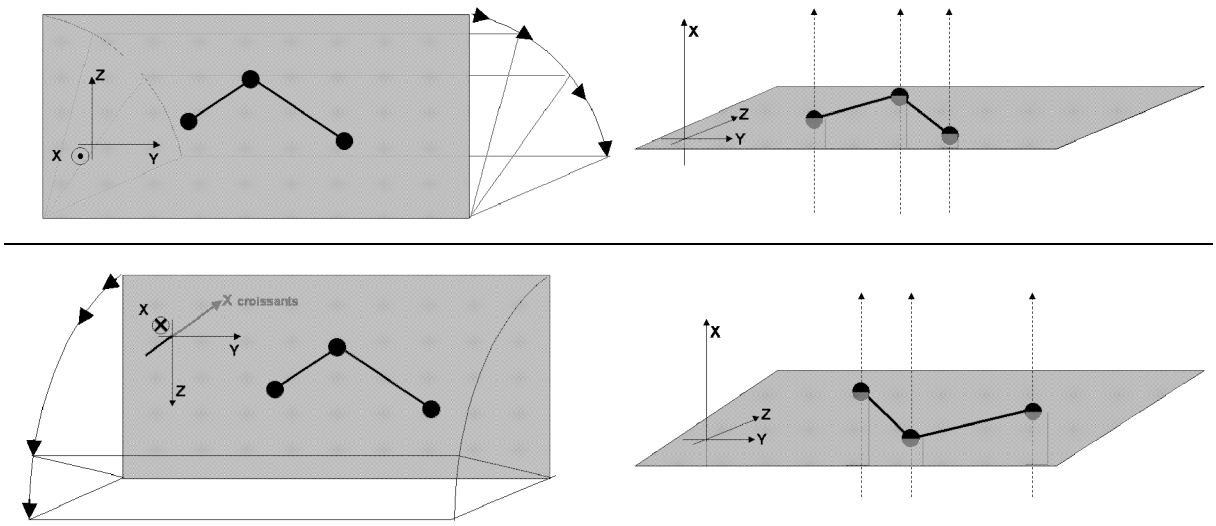


Figure 8 : deux options pour l'organisation de l'axe de simulation.
En haut à gauche : l'axe « sort » de l'écran (métaphore du tableau)
En bas à gauche, l'axe « s'enfonce » dans l'écran (métaphore du marcheur).

La *métaphore du tableau* est privilégiée dans GENESIS 1.5 ; elle nous semble être plus naturelle pour quiconque a déjà pris des notes en cours et permet que le haut de l'établi apparaisse au fond de l'espace tridimensionnel dans un mouvement de basculement minimal.

Notons, enfin, deux autres artifices de représentation qui facilitent grandement la conceptualisation de l'espace :

- D'une part, le plan de l'établi est matérialisé dans la fenêtre de simulation par le représentation des grilles qui quadrillent l'établi.
- D'autre part ce même plan est opaque (ou translucide) de telle sorte qu'on puisse aisément différencier les modules qui sont derrière lui (i.e. : dont la position X est négative) de ceux qui sont devant.

Navigation et interaction avec l'établi

Les principes mis en œuvre pour la navigation par manipulation directe dans l'espace tridimensionnel sont relativement courants[1]. Une palette d'outil permet de sélectionner le type de transformation à appliquer à la vue. Les transformations se font pour l'essentiel en *tirant* la souris sur la représentation.

Toutefois il faut noter que le zoom est particulier : il n'agit *que* sur les axes Y et Z du plan de l'établi. Un autre outil (amplitude) permet de *dilater* indépendamment l'axe X de simulation pour révéler les mouvements de faible amplitude. Le découplage ainsi obtenu entre l'axe des mouvements et l'espace topologique permet, en quelque sorte, de séparer les problèmes : le choix de la portion de l'établi à afficher pour révéler ou cacher la hiérarchie des sous-objets et celui de l'amplitude caractéristique des mouvements à observer sont indépendants[2].

GENESIS tire partie de la cohabitation de l'établi et de l'espace de représentation 3D. L'utilisateur, en effet, a la possibilité de *lier* les taux de zoom et positionnement des deux espaces (dans les préférences utilisateurs). Dès lors, il peut comme il en a l'habitude désigner la portion de son objet qui l'intéresse *dans l'établi* ; GENESIS calcule alors le taux de zoom et la position de la caméra dans la fenêtre Regarder qui assurent, compte tenu de l'angle d'incidence, que la portion visible de l'établi soient la même dans les deux représentations. Ainsi, il est possible de bénéficier des facilités qu'offre la représentation plane.

Dans le même ordre d'idée, la *sélection courante* est repérée dans la fenêtre de simulation. En sélectionnant dans l'établi le(s) module(s) qui l'intéresse(nt), l'utilisateur est ainsi mieux à même de les repérer dans la fenêtre 3D pour suivre leurs évolutions.

Enfin, nous noterons que toute modification automatique du point de vue (initialisation, vue établi, vue de coté…) est réalisée progressivement par une interpolation entre la vue d'origine et la vue finale. Notamment, avant qu'une simulation ne soit lancée, la vue affichée dans la simulation est d'abord identique à celle de l'établi (vue de face) puis bascule vers la vue préférentielle de l'utilisateur. Ainsi, ce dernier accompagne le changement d'angle de vue et comprend plus aisément le point de vue final.

[1] Parmi les deux options traditionnelles nous avons privilégié l'idée que c'est l'espace qui est déplacé, et non la caméra. Notons, incidemment, que les *rotations* doivent encore être améliorées. 3 outils coexistent à ce jour pour tourner l'objet autour des axes X (mouvements) Y et Z (établi) ; il conviendrait de les remplacer par un outil unique qui permettrait de « saisir » l'objet en un point pour travailler l'angle de vue.

[2] La décision de séparer l'amplitude et le zoom doit encore être validée – ou invalidée – par l'expérience.

Organisation de la fenêtre et manipulation par widget

La figure 9 présente le module Regarder de la fenêtre de simulation et montre comment les principales fonctionnalités sont accessibles.

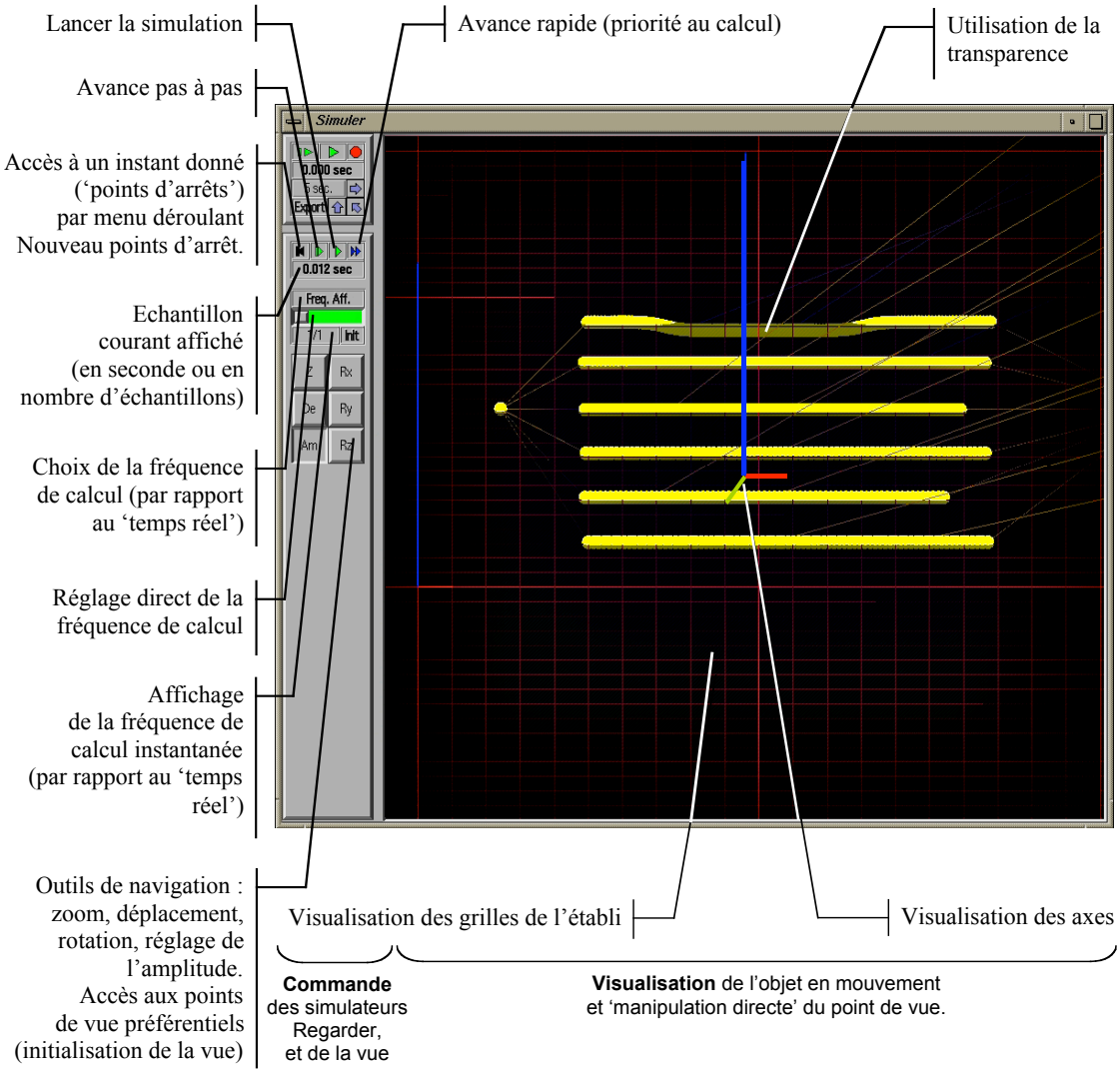


Figure 9 : la fonctionnalité Regarder dans la fenêtre de simulation (GENESIS 1.5)

Aux fonctionnalités accessibles par *widget* ou par manipulation directe de l'espace de représentation (pour la navigation) qui apparaissent sur cette figure s'ajoutent, dans la fenêtre des préférences utilisateurs, quelques options de visualisation :

- La possibilité de n'afficher que la partie de l'objet qui est *sélectionnée* sur l'établi ou au contraire de toujours afficher la totalité de l'objet (comme indiqué plus haut, les modules sélectionnés sont alors dessinés d'une façon particulière qui permet de les différencier). L'affichage exclusif de la sélection *clarifie* l'écran et de *fait ressortir* la partie de l'objet à laquelle s'intéresse l'utilisateur.
- La possibilité de ne pas afficher les modules <MAT> (leur affichage est coûteux et surchargent parfois le graphique de façon malvenue).
- La possibilité d'afficher (resp. de cacher) la grille de l'établi et les axes X, Y, Z ; l'activation / désactivation de l'opacité de l'établi.

Enfin, signalons que le pas de simulation couramment affiché dans le module Regarder est matérialisé sur la représentation du signal par une *barre de défilement* (cela suppose, bien sûr, que ces deux modules sont simultanément ouverts – voir figure 6 p.317, bas droite). On bénéficie, ainsi, d'une mise en correspondance de l'affichage *spatial* de l'objet et de l'affichage *temporel* du signal.

Points d'arrêt

Le noyau fonctionnel des simulateurs comprend une fonction qui consigne l'état courant de l'objet simulé (position et position retardées de chacun de ses <MAT>) dans un fichier *état* au format .stt[1]. Une seconde fonction permet, de façon réciproque, de redémarrer la simulation dans une configuration enregistrée dans un tel fichier – il s'agit, en quelques sortes, d'initialiser l'objet avec des conditions initiales décalées dans le temps.

L'instant (en seconde ou nombre d'échantillon) correspondant à un tel fichier état est nommé *points d'arrêt*. Le principe des points d'arrêts offre à l'utilisateur la possibilité d'accéder à un instant donné de l'histoire de l'objet, sans avoir à reprendre la simulation depuis l'instant initial t=0[2]. Il est extrêmement utile pour visualiser plusieurs fois de suite le même phénomène ou pour accéder rapidement à une date tardive.

L'accès à l'un des points d'arrêt existant pour l'objet se fait, dans la fenêtre de simulation, à l'aide d'un menu. Deux procédés sont disponibles pour générer un (ou des) point(s) d'arrêt :

- L'utilisateur peut d'abord demander (dans le fenêtre des préférences) qu'un point d'arrêt soit créé à période fixe (toutes les secondes de l'histoire de l'objet, toutes les 10 secondes, etc.). Ainsi, une succession de « photographies » est constituée. Plus tard, ces points d'arrêt permettent d'accéder aisément à une date quelconque (il suffit de choisir le point d'arrêt qui la précède).
- Il est également possible de demander la création d'un point d'arrêt à une date spécifique. Ceci fait, GENESIS passe en mode de simulation accélérée (i.e. : la priorité est donnée au calcul de simulation et la fréquence de rafraîchissement de l'affichage est diminuée). Lorsque la date spécifiée est atteinte, le simulation est stoppée, un fichier d'état .stt est généré et la date est ajoutée à la liste des points d'arrêt.

Enfin, un point d'arrêt et un fichier état *.stt* sont systématiquement générés à l'issue de chaque simulation Ecouter afin que le calcul du son puisse être poursuivi ultérieurement, si l'utilisateur le désire. La poursuite d'une simulation peut indifféremment avoir lieu dans la même cession de travail ou dans une cession ultérieure[3].

---

[1] Voir partie IV chapitre 2.

[2] A la condition, bien sûr, qu'une simulation antérieure ait déjà atteint l'instant considéré.

[3] Cette dernière option n'est pas disponible dans la version 1.51, dans la mesure ou les fichiers *.top* ne conservent pas la liste des points d'arrêt. Elle sera implémentée prochainement.

20.4 - Notes sur l'implémentation

Les fonctionnalités de simulation sont parmi celles qui ont nécessité un travail technique conséquent. Ce paragraphe propose un descriptif rapide des procédés mis en œuvre.

Dans GENESIS 1.5, toute simulation est effectuée par un processus Simulateur indépendant (noyau fonctionnel). Chaque programme Simulateur sait : lire un fichier .top, appeler les fonctions d'allocation, d'initialisation et de calcul de la bibliothèque CORDIS-OFF (moteur de simulation), échanger, enfin, des données avec GENESIS. L'indépendance des Simulateurs relativement à GENESIS assure d'une part que les calculs de simulation soient optimisés et d'autre part que l'interface soit fluide[1].

A chaque établi ouvert peut correspondre deux simulateurs indépendants : un simulateur Ecouter pour le calcul du signal sonore et un simulateur Regarder pour la visualisation de l'objet en mouvement. Les simulateurs Ecouter calculent *aussi vite que possible* les échantillons successifs, jusqu'à atteindre la date spécifiée. La fréquence de calcul des simulateurs Regarder est par contre contrôlée à tout instant par GENESIS[2]. Les informations concernant l'établi actif sont portées dans la fenêtre de simulation. Les simulateurs qui correspondent à des établis non-actifs tournent, eux, en tâche de fond et n'ont que peu d'échanges avec GENESIS.

On peut distinguer plusieurs *états* pour les programmes Simulateurs de chaque établi suivant qu'ils supportent une action Regarder ou Ecouter, qu'ils sont créés ou non (la création du processus Simulateur n'intervient que lorsque c'est nécessaire ; de même, le processus est détruit lorsque l'utilisateur stoppe la simulation), qu'ils sont en pause ou en cours de calcul, qu'ils correspondent à l'établi actif ou non. Pour chacun de ces états, le débit et la qualité des informations échangées entre GENESIS et les Simulateurs sont spécifiques.

Quatre procédés sont utilisés pour assurer une communication modulaire entre les différents processus (tableau de la figure 10) :

- *Des fichiers.* Un fichier .top décrivant l'objet est utilisé à la création du simulateur. Par ailleurs, des fichiers son au format flottant .echf[3] sont générés par les simulateurs Ecouter. Ils sont lus par GENESIS pour afficher le signal.
- *Une file de messages.* De façon générale, GENESIS et les simulateurs communiquent d'abord par messages. Les messages sont utilisés à la manière d'une télécommande pour transmettre des ordres généraux. Des *Threads* (processus légers) sont utilisés dans chaque programme pour assurer la réception des messages sans que les calculs du processus père ne soient ralentis.
- *Un segment de mémoire partagé.* La mémoire partagée est utilisée pour la communication entre GENESIS et les simulateurs Ecouter et Regarder qui correspondent à l'objet actif (i.e. : celui dont l'établi a été sélectionné par l'utilisateur, et dont les données de simulation doivent être portées dans la fenêtre de simulation). La

[1] La simulation est, dans GENESIS, l'opération la plus coûteuse. La complexité des calculs nécessaires pour simuler l'objet n'est guère compressible (la bibliothèque CORDIS-OFF a été écrite avec un souci d'optimisation). Par contre, *l'environnement* des calculs (interface, communication, gestion des appels à la libraire CORDIS-OFF) nécessitait d'être optimisé avec attention.

[2] Elle devient cependant maximale dans le cas de l'avance rapide.

[3] Voir la partie IV, chap.2 pour la description des formats de fichier.

mémoire partagée autorise un flux de données rapide et important. Elle est structurée en deux parties : d'une part une en-tête pour l'échange d'ordres (nombre de pas à calculer, pause, etc.), d'autre part un tableau. Ce tableau est renseigné par le simulateur Regarder avec l'état courant de l'objet (position de chaque <MAT> sur l'axe de simulation). Il est lu par GENESIS pour afficher l'état dans l'espace Regarder de la fenêtre de simulation (i.e. : pour dessiner l'objet en mouvement).

- *Plusieurs sémaphores,* enfin. Ils règlent d'une part les accès concurrents à la mémoire partagée et permettent d'autre part le contrôle de la fréquence de calcul du simulateur Regarder de l'établi actif (synchronisation).

	Type de simulateur	Non créé	En pause	En cours de calcul
Etabli Actif	Simulateur Regarder	-	Message (bloquant) + mémoire partagée	Message (réception dans un thread) + Mémoire partagée + sémaphore
	Simulateur Ecouter	-	Message (bloquant) + Mémoire partagée	Message (réception dans un thread) + Mémoire partagée + sémaphores
Etabli non-actif	Simulateur Regarder	-	Message (bloquant)	Message (réception dans un thread)
	Simulateur Ecouter	-	Message (bloquant)	Message (réception dans un thread)

Figure 10 : répartition des procédés de communication suivant l'état du simulateur

Enfin, pour compléter cette présentation signalons :

- Que la lecture du son est effectuée par un processus léger (*thread*) de GENESIS.
- Qu'un autre processus léger est utilisé pour assurer les calculs d'enveloppe des signaux sonores[1].

La figure 11 page suivante synthétise l'architecture des fonctionnalités de simulation que nous avons mise en œuvre dans GENESIS 1.5.

[1] Incidemment, les tests ont démontré que cette solution est insuffisante. Comme le font la plupart des logiciels d'édition de sons, il conviendrait à l'avenir de définir un format de fichier pour stocker les enveloppes, plutôt que d'avoir à les recalculer à chaque changement du taux de zoom temporel dans le module Signal de la fenêtre de simulation.

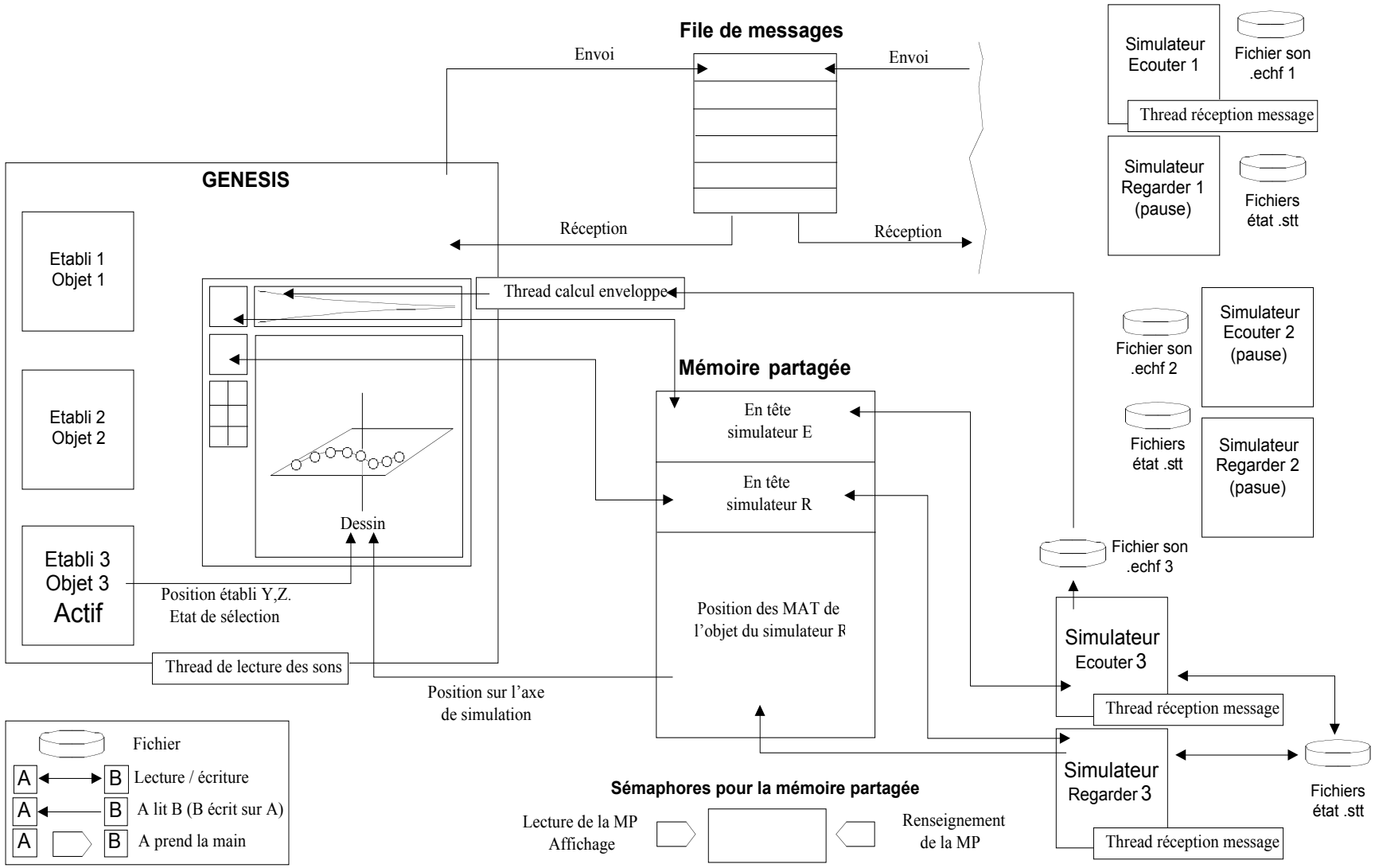


Figure 11 : communication entre GENESIS et les Simulateurs

Chapitre 21

PERSPECTIVES POUR LES FONCTIONNALITES DE SIMULATION

La fenêtre de simulation est de conception récente et son implémentation doit être complétée. Nous exposons ci-après deux types de développements, à court terme et pour le plus long terme, qui ensemble permettront de répondre à toutes les tâches identifiées.

21.1 - Finalisation

A court terme, il faudra mener une *évaluation* exhaustive des fonctionnalités et de l'ergonomie, une *incrémentation* des fonctionnalités et une *optimisation* de certains *process*.

21.1.1 - Evaluation

La fenêtre de simulation n'a pas été expérimentée *in-situ* et, en conséquence n'a pas traversé de cycle itératif. Trois points nous semblent particulièrement sensibles, et devront être évalués avec une attention particulière. Il s'agit tout d'abord des fonctionnalités de navigation et notamment du découplage que nous avons proposé entre le zoom et le contrôle de l'amplitude (le zoom n'agit que dans le plan Y, Z de l'établi). Il s'agira ensuite d'évaluer si les affichages simultanés de l'objet en mouvement et de la représentation temporelle du signal, qui renvoient chacun à une notion du temps différente, perturbent les utilisateurs. Il s'agira enfin – point crucial – d'estimer dans quelle mesure les choix effectués facilitent, comme nous l'espérions, la compréhension de l'articulation entre l'établi et l'espace tridimensionnel de l'objet en mouvement.

21.1.2 - Incrémentation

Un certain nombre des tâches identifiées au paragraphe 20.2 et dont les fonctionnalités sont d'ores et déjà spécifiées ne sont pas implémentées dans la version actuelle de la fenêtre de simulation. Leur mise en œuvre permettra par exemple de *diminuer la durée d'un son*, de *choisir, parmi les voies du fichier son, celles qui doivent être visualisées et écoutées* ou encore de *choisir des options de visualisation évoluées* (lumière, texture, etc.), ou encore de conserver les fichiers d'état .stt et la liste des points d'arrêt dans les fichiers .top pour une utilisation ultérieure.

Deux tâches nécessitent encore réflexion.

Tout d'abord, il devrait être possible de ne simuler qu'une *partie* d'un objet. Ici, deux options ergonomiques sont envisageables : introduire un *mode* dans l'actuelle fenêtre de simulation (de telle sorte qu'elle affiche soit les phénomènes relatifs aux objets complets soit ceux qui résultent de la simulation de la sélection) ou ajouter une seconde fenêtre, identique à la première mais qui fonctionnerait exclusivement en relation avec la sélection réalisée dans l'établi (comme la fenêtre des paramètres par exemple). Le choix entre ces deux options devra être fait à partir d'une évaluation sur *IHM représentée*[1], qui considèrera notamment les différentes sous-tâches impliquées et les modèles mentaux associés.

[1] Partie II Chapitre 3.

Ensuite, GENESIS devra à terme offrir la possibilité de programmer une *campagne de simulations* enchaînées les unes à la suite des autres – pendant une nuit, par exemple – en *batch*[1]. Ce type de fonctionnalités, cependant, dépasse le simple cadre de la fenêtre de simulation. Il appelle à ce que le mode Gestion, dont nous avons discuté dans la partie IV, soit développé. Rappelons que le mode Gestion devrait qui plus est permettre une navigation plus aisée au sein des comptes utilisateurs, afin par exemple de permettre d'écouter les sons de plusieurs objets à la suite, d'exporter un ensemble de phénomènes, de supprimer certaines données, de déplacer des fichiers, etc.

21.1.3 - Optimisation

La troisième catégorie des travaux de finalisation qu'il faudra conduire sur les fonctionnalités de simulation concerne l'optimisation d'un certain nombre de *process* et particulièrement des opérations de dessin.

Lorsqu'ils n'ont pas ou peu à interagir avec GENESIS, les programmes Simulateur sont de fait particulièrement optimisés[2]. Par contre, dès lors que les flux échangés deviennent plus importants et surtout lorsque que le Simulateur doit être synchronisé avec GENESIS, ce gain devient nul.

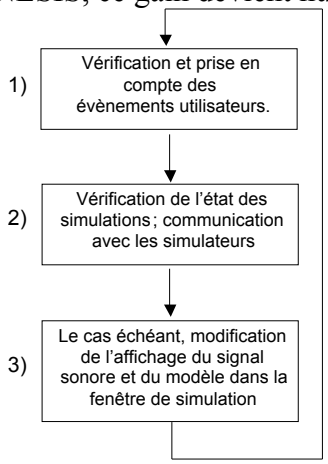


Figure 12 : affichage de l'objet en mouvement

En termes moins techniques, il s'avère qu'on a considérablement gagné en vitesse en ce qui concerne le calcul des sons, mais point en ce qui concerne l'affichage des objets en mouvement.

En fait, ce problème tient à ce que l'affichage de l'objet en mouvement est réalisé au sein même de GENESIS, au cours d'une boucle qui enchaîne le traitement des évènements utilisateurs, une phase de communication avec les simulateurs (vérification / envoi des messages), puis l'affichage (figure 12). Lorsque l'utilisateur souhaite afficher tous les échantillons calculés, le simulateur se voit obligé d'attendre la complétude de la boucle avant que de poursuivre la simulation.

Une évolution importante serait donc de déporter toutes les opérations graphiques de la fenêtre de simulation dans un nouveau programme. Plusieurs études et prototypes ont été conçus en ce sens. Des problèmes de nature technique sont apparus et devront être résolus.

Un autre point nécessite également discussion. Dans les travaux qui composent (ou assemblent) dans un même objet plusieurs structures vibrantes, chefs, instrumentistes , etc, certains des sous objets n'interviennent qu'à des moments donnés de la partition complète. A l'heure actuelle cependant, tous les sous-objets sont constamment simulés, même lorsqu'ils n'ont pas (ou très peu) d'incidence sur le reste de la structure et sur le son calculé et que, réciproquement, ils ne sont pas (ou peu) excités par sympathie.

Une évolution pertinente en termes d'optimisation consisterait donc à permettre un séquencement (ou *scheduling*) des sous-objets dans le temps de la simulation des divers sous-objets. L'utilisateur spécifierait explicitement quand simuler telle ou telle partie de

[1] i.e. : en l'absence de toute interaction.

[2] On observe alors, suivant les modèles, un gain de temps de calcul de 20 à 30 % par rapport à la version 1.4.

l'objet. Du fait de l'augmentation des exigences des utilisateurs vis-à-vis du logiciel, elle-même consécutive à son développement récent, le séquencement apparaît aujourd'hui *nécessaire*.

Sa mise en œuvre, cependant, n'est pas sans difficulté. Elle suppose que le temps devienne un paramètre explicite et contrôlable par l'utilisateur pendant la phase de lutherie, alors même qu'à l'heure actuelle il n'apparaît que dans la représentation du signal, c'est-à-dire du côté des phénomènes ou produits.

Lors de l'analyse des modes de GENESIS et de leurs fonctionnalités respectives (partie IV, chapitre 11), nous avons montré comment la dimension temporelle devrait intervenir explicitement au niveau des modes Jeu et Composition. Une fonctionnalité permettant un séquencement des sous-objets devrait, ainsi, s'insérer dans le mode Composition. Le séquencement appelle donc, aux côtés d'autres fonctionnalités, à ce que ce mode soit développé.

21.2 - Métrologie

Lors de l'analyse des tâches nous avons rappelé que la *mesure* est l'une des modalités nécessaires pour l'observation des phénomènes. Dans GENESIS 1.5, la seule mesure possible est celle de l'amplitude du signal sonore. Elle est insuffisante. La première perspective à moyen terme que nous retenons quant aux fonctionnalités de simulation concerne le développement de la métrologie.

La réflexion, en la matière, ne fait que débuter. Dans ce court paragraphe, nous résumons les principaux enjeux pointés par la métrologie.

La *métrologie* est la mesure des phénomènes. On peut distinguer deux cas suivant que la mesure est faite *pendant* la simulation ou *après* la simulation. Dans le premier cas, il s'agit de mesurer des flux. Dans le second, il s'agit d'observer ou *d'analyser* des données enregistrées dans un fichier. Ici, il est nécessaire qu'un *autre* logiciel intervienne : GENESIS ne saurait en effet devenir un environnement complet pour l'analyse des signaux. Il n'est pas envisageable, ni même souhaitable d'y implémenter les nombreuses fonctionnalités qu'offrent de tels environnements. Ainsi, la métrologie repose, pour partie, sur la possibilité *d'exporter* diverses données phénoménologiques à destination de logiciels tiers.

A quel endroit devrons nous alors mettre la limite ? Quelles fonctionnalités doivent être intégrées à GENESIS ? Nous proposons trois éléments de réponse à ces questions.

Tout d'abord, les mesures *en flux* de l'objet en cours de simulation (position de tel ou tel <MAT>, force exercée par tel ou tel <LIA>, amplitude moyenne des oscillations d'un sous-objet, etc.) ne peuvent être envisagées qu'au sein de GENESIS[1]. Ici, une piste intéressante consisterait à ce qu'il soit possible de modifier certains paramètres dans le temps même de la simulation pour faire converger la *mesure* d'une certaine propriété (mettons : l'amplitude des mouvements d'un <MAT> sous l'effet d'une interaction) vers une valeur souhaitée. Ce type de fonctionnalités de mesure devrait pouvoir être présenté comme un mode supplémentaire de la fenêtre de simulation, aux côtés de la visualisation de l'objet en mouvement et du signal sonore.

[1] A ce sujet, rappelons que la base des modules d'entrée sortie définis par CORDIS-ANIMA et utilisée dans GENESIS n'est pas complète (partie IV chapitre 1). Une refonte des principes de la mesure au sein des objets est à l'étude au laboratoire. Elle devrait aboutir à ce que *tout* signal échangé entre modules puisse être capté sans perturber l'objet (mesure « non perturbante »).

Ensuite, il convient de développer une analyse précise de l'ensemble des fonctionnalités métrologiques essentielles dans le contexte de GENESIS. Les types de mesures et d'analyse que l'utilisateur a *besoin* d'opérer de façon *récurrente* devront être rendues possibles dans GENESIS. Parmi celles-ci figurent, par exemple, les mesures qui ont trait au LNL. La conception d'une interaction non linéaire à l'aide d'un LNL est en effet souvent une opération complexe. Elle est rendue plus difficile encore du fait qu'il n'est pas possible d'ausculter le comportement du module – par exemple de visualiser la plage de la caractéristique sur laquelle on se situe[1].

Enfin, plusieurs travaux ont montré que le paradigme du modèle physique, et particulièrement CORDIS-ANIMA, présentent en eux-même un potentiel certain pour la métrologie. Par exemple, il est possible d'opérer une décomposition temps fréquence *auto-adaptative* (i.e. : dans laquelle la précision temporelle dépend de la fréquence analysée à la manière des *ondelettes*) à l'aide d'un *banc d'oscillateurs* sur lequel le signal à mesurer est décomposé [Incerti.96], [Incerti&Cadoz.95a].

Dans un tel *fréquencemètre modal*, chacun des oscillateurs entre plus ou moins en résonance en fonction du signal qui lui est appliqué. Pour peu que les fréquences de résonance et coefficients d'amortissement soient choisis de façon judicieuse, l'ensemble des amplitudes des oscillateurs à un instant donné constitue une « mesure » du spectre du signal.

Le *fréquencemètre modal* n'est qu'un exemple des possibilités que laisse entrevoir le paradigme CORDIS-ANIMA en matière de métrologie. Les recherches sur ce sujet se poursuivent. Elles font espérer qu'il soit à terme possible de proposer, au sein de GENESIS, une approche de la problématique de la mesure à la fois originale et cohérente avec l'ensemble de la démarche.

21.3 - Vers une véritable multisensorialité ; connexion au moteur temps-réel TELLURIS

La troisième et dernière perspective que nous évoquerons concerne la connexion de GENESIS au moteur de simulation temps-réel TELLURIS et au périphérique à retour d'effort TGR©. Il n'est pas nécessaire de discuter ici des nombreuses attentes entretenues relativement à cette possibilité. Compte tenu de ce que le paradigme de la Simulation Interactive et Multisensorielle d'Objets Physiques est l'une des motivations essentielles de nos travaux, la connexion au moteur temps réel apparaît à la fois comme un *objectif* et un *aboutissement.*

Les évolutions récentes de TELLURIS, auxquelles nous avons participé, ont conduit à mettre en œuvre une architecture de type client-serveur. Le serveur TELLURIS tourne exclusivement sur une machine *Power Challenge* (SGI) multiprocesseur. Il permet de procéder aux simulations multisensorielles interactives des objets CORDIS-ANIMA et assure les échanges avec les périphériques. Plusieurs clients (GENESIS, MIMESIS[2] ou d'autres environnements) se connectent à TELLURIS et commandent à distance les simulations par des requêtes.

---

[1] Une version ad hoc de GENESIS, développée à l'occasion d'un travail sur le frottement d'archet, offre quelques fonctionnalités élémentaires en la matière. Ce travail peut servir de base à une analyse des tâches en ce qu'il a mis en évidence quelques *mesures* nécessaires et visualisations graphiques pertinentes.

[2] Rappelons que MIMESIS est l'environnement dédié à la création d'images animées et aux Arts du Mouvement qui est développé au sein du laboratoire.

Une version expérimentale de GENESIS a d'ores et déjà été produite. Cette version avait essentiellement pour objectif d'aider les développeurs de TELLURIS dans leurs travaux. Sa conception a cependant mis en lumière les principaux points qu'il nous faudra considérer pour, dans un avenir proche, mettre en œuvre une connexion plus pertinente.

Outre le développement de la partie client, la connexion de GENESIS au moteur temps réel nécessite l'ajout que de trois fonctionnalités :

- Une télécommande pour TELLURIS, qui permette le contrôle des simulations (charger un modèle, lancer ou arrêter les simulations, etc.) et le choix des options du moteur (coefficient de normalisation à appliquer aux signaux entrant et sortant, options de visualisation, etc.). L'ergonomie de cette télécommande pourra être similaire à la partie *commande* de l'actuelle fenêtre de simulation temps différé (principe de cohérence).
- Une fonctionnalité permettant de spécifier comment l'objet doit être réparti sur les différents processeurs de la machine. Les Ensembles sont tout indiqués pour être à la base d'une telle fonctionnalité. Sa conception ne pose aucune difficulté particulière.
- Des fonctionnalités pour la modification des paramètres en cours de simulation, support d'un possible *accordage dynamique* des objets[1] (affectation, multiplication, etc.). Ces fonctionnalités ont été récemment ajoutées du côté du serveur TELLURIS. Etant prévues, elles ont été prises en compte pour les choix architecturaux du client GENESIS et leur implémentation finale ne pose, à nouveau, pas de difficulté importante.

Un autre problème auquel il faudra réfléchir concerne la façon dont les simulations temps réel et temps différé, qui rappelons-le sont complémentaires et doivent à terme cohabiter, peuvent s'articuler dans GENESIS. Ce problème de nature ergonomique figure parmi les plus difficile qu'il nous faudra résoudre ici.

Enfin, il est certain que la connexion de GENESIS au simulateur TELLURIS nécessite que celui-ci soit finalisé. Il faudra donc continuer à travailler sur les questions techniques très délicates que pose le temps réel dur *et* le périphérique à retour d'effort. Il faudra, de plus, envisager la portabilité du moteur sur d'autres machines. Ces problèmes, cependant, dépassent le cadre de nos propres travaux.

De façon générale, l'architecture de l'environnement GENESIS tient compte de la connexion au moteur temps réel, qui était un objectif tacite pour l'ensemble de nos travaux. Ainsi, celle-ci devrait pouvoir être opérationnelle à très court terme.

De plus, comme nous l'indiquions dans la partie IV, l'essentiel des fonctionnalités de GENESIS pour la lutherie des objets CORDIS-ANIMA ne dépend en rien du moteur de simulation utilisé. Ainsi, bientôt, il sera possible au sein de GENESIS de confronter à l'expérience le paradigme de la Simulation Multisensorielle et Interactive d'Objets Physiques telle que nous l'avons introduit dans les premières parties de ce rapport.

[1] Partie III chapitre 8.

Conclusions
et perspectives

Notre travail est une contribution au programme de recherche de l'ACROE dont l'objectif général est le développement de concepts et d'outils pour la création artistique dans le cadre de l'informatique et de l'ordinateur. La création musicale, dans ce contexte, est abordée comme un processus qui se développe autour d'une situation considérée comme centrale, la relation instrumentale (dans ce qu'elle a d'interactif et de multisensoriel). Avec l'ordinateur, cette relation se réalise non plus avec des instruments réels, mais avec des « simulacres », des représentations virtuelles - réalistes ou non - de la réalité des dispositifs physiques producteurs de son. Ceci introduit une nouvelle dimension dans le processus de création : créer des représentations d'objets instrumentaux à l'aide d'objets virtuels. Le processus de création complet se place alors dans une nouvelle perspective où trois modes complémentaires s'articulent, qui peuvent être supportés par un seul outil : créer des instruments virtuels, en jouer et composer. Le premier joue un rôle fondamental et on peut dire qu'à travers lui, il y a un changement de paradigme. L'ordinateur n'est plus un lieu de synthèse des sons, mais un outil de représentation des « corps sonores » (au sens de Schaeffer).

L'objectif de notre travail était, étant donnés le langage de modélisation et simulation CORDIS-ANIMA créé par le laboratoire, le paradigme général de la Simulation Multisensorielle Interactive d'Objets Physiques, et l'ébauche de premières interfaces utilisateurs pour leurs mises en œuvre en situation de création musicale, de concevoir un environnement complet à même de promouvoir et soutenir un processus de création musicale explorant et exploitant tout le potentiel de cette approche.

La version 1.5 du logiciel GENESIS, qui est la concrétisation actuelle de ce travail, se présente comme un outil significatif, aux dires des utilisateurs de plus en plus nombreux qui l'ont abordé ou qui ont d'ores et déjà réalisé des œuvres avec les versions qui l'ont précédée. A travers sa conception et son développement, nous avons été amenés à nous poser nombre de problèmes de diverses natures, relatifs à l'informatique pour la création artistique (en particulier la musique), au processus de création lui-même et, bien sûr aux interfaces homme-ordinateur. Nous tentons, dans cette conclusion, avec un recul encore bien modeste, de dégager l'essentiel des résultats que nous pensons avoir mis au jour.

Rappelons tout d'abord la singularité de la problématique qui se posait. La création artistique n'est en effet pas une « tâche ordinaire » : d'une manière générale, si une œuvre se définit souvent à travers un projet initial, celui-ci n'est jamais complètement arrêté avant l'activité même de sa réalisation. L'activité de création artistique n'échappe pas à l'analyse mais résiste en revanche à la standardisation qui caractérise bien souvent le travail du développeur de logiciels. La conception d'une interface, par ailleurs, a en général pour objet de permettre la réalisation d'une tâche *connue* et *spécifiée*. Mais l'outil pour la création artistique, en tout cas celui que nous voulions concevoir, ne saurait s'identifier à un « outil de production » et échappe en conséquence à cette problématique.

La « création d'une interface pour la création » pose alors des problèmes spécifiques et originaux par rapport à tout autre type d'interface. Il serait notamment vain de vouloir spécifier *a priori* la nature des tâches de l'utilisateur. Tout processus de création s'invente et se réinvente constamment au contact de l'outil, dans la médiation qu'impose ce dernier. Mais, à l'inverse, la conception de l'outil ne peut être conduite qu'en connaissant le processus de création qu'il est destiné à permettre – les *tâches* en termes d'IHM.

Certes, l'interface doit être agréable, rapide, sympathique, etc. Mais cela ne suffit pas, loin s'en faut. Il a fallu en particulier, au-delà de la construction d'un outil *utile* et *utilisable*, tenter de mettre en évidence ce qui peut malgré tout, dans le processus de création, être explicité et formalisé, dégager ce qui répond à des schémas stables et

généralisables qui puissent être traduits dans des fonctionnalités, avec une ergonomie. La difficulté est dans ce cas de trouver un juste compromis entre un cadrage qui ne stérilise, ne ferme pas le processus de création, ne le compromette pas ou ne le renvoie pas dans la banalité, le trivial, le stéréotypé et une ouverture qui ne le rende pas non plus inopérant par trop de généralité.

Cette situation a tout d'abord posé des problèmes d'ordre méthodologique.

La conception d'une interface pour la création musicale nécessite, certes, une bonne maîtrise des bases conceptuelles de l'IHM et une connaissance précise des principes essentiels qui font une « bonne » ergonomie. Nous avons alors effectué une large excursion dans le domaine de l'IHM (résumée dans la troisième partie du mémoire). Mais la conception d'un environnement destiné au créateur *ne peut* se ramener aux seuls aspects de l'IHM.

Nous avons dû tenter de caractériser le potentiel de renouveau du processus de création porté par la Simulation Multisensorielle (SIMOP) en nous appuyant sur une étude bibliographique centrée sur l'impact qu'ont eu, dans le demi-siècle précédent, les avancées technologiques sur le développement de nouveaux instruments et finalement sur le processus de création musicale. A cette analyse a succédé une phase de synthèse orientant nos choix des fonctionnalités essentielles et de l'ergonomie de GENESIS.

Mais la conception d'une telle interface demande également la mise en œuvre d'une méthodologie rigoureuse. Nous avons ainsi pu mettre en évidence la nécessité d'une méthodologie procédant par incréments et par rebouclages, basée sur un aller-retour entre la spécification, la conception, le prototypage et l'évaluation. Il faut être conscient qu'une telle approche présente les risques d'une dispersion et de la perte des objectifs à atteindre. Nous l'avons cependant pratiquée résolument et sommes convaincus qu'en dessous de cette exigence, le projet aurait été rapidement compromis.

Par ailleurs, nous avons eu recours à un processus de maturation nourri d'une collaboration constante avec les utilisateurs. Cette collaboration ne s'est pas limitée aux aspects ergonomiques (pour permettre par exemple de rendre le logiciel plus utilisable ou efficace) ni à la compréhension des problèmes et des tâches des utilisateurs. Nous avons pu nous rendre compte que bien souvent les utilisateurs, nourris et formés au contact des outils plus anciens, notamment informatiques, souhaitaient retrouver des principes et fonctionnalités auxquels ils étaient habitués. La collaboration avec les utilisateurs présente en fait également un danger de dispersion et de dissolution des objectifs, auquel il nous a fallu résister.

En fait, la collaboration avec les utilisateurs s'est déroulée comme une participation active à la « création de l'outil de création ». Autour de quelques objectifs communs – centrés sur une vue très générale du paradigme de la Simulation Interactive et Multisensorielle d'Objets Physiques et de la démarche qu'on souhaite autour de CORDIS-ANIMA – il s'est agit en fait, dans une *interaction* mêlant les trois types de points de vue, scientifique, technologique et artistique, d'inventer, de créer le processus de création. L'outil, alors, apparaît à la fois comme le moyen qui permet de le mettre en œuvre et d'une certaine façon comme sa représentation.

S'il nous faut défendre un premier résultat de ces travaux, nous avançons qu'il réside peut-être dans cette démarche même de « création » du logiciel, dans laquelle sa « raison d'être » s'est inventée et confirmée au fur et à mesure de sa conception. Nous souhaitons vivement que ce processus ne s'arrête pas là et qu'il se prolonge dans les nombreux développements futurs qui nous sont d'ores et déjà apparus comme nécessaires ou souhaitables.

A l'issue de ce travail, même si nous avons dû renoncer à certaines dimensions et s'il appelle d'autres développements et de nouvelles expériences, nous disposons d'un outil opérationnel. Limité par le choix fondamental du temps différé (du moins dans un premier temps) et à la projection *topologique* du langage CORDIS-ANIMA, il propose :

- Une architecture générale qui place l'objet CORDIS-ANIMA (ou plutôt sa représentation) au *centre* du logiciel et au centre des tâches de l'utilisateur.
- Une manipulation directe des modèles CORDIS-ANIMA, de telle sorte que représentation et possibilités d'interaction donnent à imaginer le modèle non pas seulement comme une construction symbolique mais comme un *objet*.
- Un appui sur le *niveau des modules*, de telle sorte que *chaque* module élémentaire CORDIS-ANIMA puisse être expérimenté et que la phase de lutherie résulte d'un processus d'assemblage.
- Un ensemble de fonctionnalités pour l'édition des paramètres.
- La possibilité de mettre en œuvre un grand nombre de modules élémentaires dans des constructions toujours plus complexes.
- Un ensemble de fonctionnalités *macro-modulaires* permettant d'expliciter les différents sous-objets et plus généralement l'existence de relations structurelles entre modules et/ou sous objets.
- Un ensemble de fonctionnalités pour *l'analyse* et la *génération* des objets CORDIS-ANIMA, fonctionnalités qui, dans les cas rares ou cela est souhaitable, construisent un *pont* entre l'univers physique des objets et celui de leurs propriétés.
- Enfin, un ensemble de fonctionnalités permettent la simulation et l'observation des *phénomènes.*

Le rôle central du *mode lutherie* et de la phase du processus de création qui le caractérise, c'est-à-dire des outils de création du modèle, concrétise le concept *d'objet physique virtuel à vocation musicale*. Nous lui avons donc consacré une part importante de nos travaux, en tentant d'apporter plusieurs solutions nouvelles aussi bien dans les fonctionnalités retenues que dans leur ergonomie. La même approche nous à permis des avancées que nous pensons significatives sur les aspects fonctionnels évolués de l'atelier et sur les modes de simulation.

Enfin, il nous apparaît que le souci du détail et de la cohérence auquel nous avons consacré une attention particulière était une nécessité. Il se passe en effet un phénomène significatif, observable directement dans le comportement et les résultats des utilisateurs : à l'élimination de toutes les petites incohérences et à la systématisation des chartes de représentation et des protocoles d'action sur l'ensemble du logiciel, fait suite une augmentation spectaculaire du dynamisme de l'utilisateur dans ses expériences, ses heuristiques et… ses trouvailles. Ceci nous fait dire, toujours dans le cadre de l'ingénierie des interfaces, que si une bonne analyse, une bonne conceptualisation, une bonne méthodologie de développement sont absolument indispensables, elles ne constituent que la moitié du chemin. L'autre tient dans une rigueur poussée et presque « esthétique » de leur réalisation concrète.

Perspectives

La version 1.5 de GENESIS est appelée, comme les précédentes, à être passée au crible d'une analyse critique impliquant, à nouveau, une observation d'utilisateurs et un *dialogue* avec ceux-ci. Cette analyse guidera la conception de nouvelles fonctionnalités et des évolutions ergonomiques. Plus, elle permettra à nouveau de réinventer le processus de création, de faire apparaître de nouvelles démarches, de nouvelles tâches.

On peut d'ores et déjà évoquer quelques perspectives immédiates, de l'ordre de la *finalisation*. Certaines ont été décrites dans le document ; nous ne rappelons ici que les principales, celles dont l'apparition devrait ouvrir de nouvelles possibilités en matière de processus de création :

- L'introduction d'une *trans-représentation* des objets, basée sur un langage. En termes de style d'interaction, une telle trans-représentation devrait permettre de pallier certains problèmes de la manipulation directe en permettant, par exemple, certaines actions répétitives telle l'assemblage d'une structure à l'aide de boucles. Mais l'essentiel est ici que l'utilisateur puisse modifier son rapport aux objets conçus en les approchant de façon plus symbolique ou abstraite.
- La conception du *système de relations et de meta-paramètres*. L'objectif est ici de rendre explicite dans l'interface le savoir-faire en matière de paramètres. Ce n'est pas seulement un facilité supplémentaire, mais bien une fonctionnalité en soi permettant de s'intéresser à des propriétés émergentes complexes des modèles liées à des relations entre leurs paramètres, au sein d'un même objet ou entre plusieurs objets, et de traiter l'édition de ceux-ci à un plus haut niveau. Ce champ n'est cependant pas sans poser quelques problèmes difficiles.
- Le développement de la notion de *Capsule*, qui devra permettre un changement d'échelle dans l'approche des objets.
- Le développement des fonctionnalités de *métrologie* qui permettront, par la mesure des phénomènes, de mieux comprendre et mieux maîtriser ce qui les détermine.
- L'ouverture, enfin, au temps réel.

A plus long terme, le dernier point est *fondamental* en ce qu'il invite à une transformation radicale de l'approche. Dans GENESIS, l'accent est mis sur des objets complexes qui, par leur structure, permettent la génération d'une forme plus ou moins complexe. Avec le temps réel et l'interaction multisensorielle, il deviendra possible de toucher, d'interagir, de sentir, de créer dans une approche véritablement instrumentale mais qui, compte tenu des libertés permises par la *simulation* devra être étudiée avec soin.

L'introduction du temps réel devra s'accompagner du développement du mode *jeu*. Ici, un nouveau champ de création devrait s'ouvrir encore. Le mode jeu et l'édition des traces gestuelles, en référence à la perception et/ou en référence à la cause, ouvrent eux aussi de nouvelles pistes d'études et de nouvelles perspectives pour le processus de création.

Enfin, une autre perspective importante peut être l'introduction des objets multidimensionnels. Le « mode » topologique de CORDIS-ANIMA a de nombreuses vertus, dont celle en particulier de permettre une investigation très large tout en conservant une relative simplicité à la fois des modèles et de leurs représentations mentales. Mais il est certain que le multidimensionnel permet la modélisation de propriétés non réalisables en topologique et qui ont cependant une pertinence évidente, en particulier dans le domaine des phénomènes non-linéaires.

Sur le plan musical, qui reste évidemment l'objectif premier, quoi que difficile à évaluer avec aussi peu de recul, on peut évoquer ce que ce logiciel permet d'ouvrir en observant d'une part les démarches et les processus qui s'organisent autour de son utilisation et en considérant d'autre part les œuvres qu'il a d'ores et déjà permis de réaliser.

En ce qui concerne le processus mis en œuvre par les artistes, il faudrait ici leur laisser la parole, mais il est important de préciser que l'approche n'est pas nécessairement facile au premier abord. Elle nécessite au départ une assistance pédagogique soutenue, qui doit être menée avec précision et rigueur et qui ne s'effectue pas en quelques minutes, comme dans les systèmes *clé en main* et prêts à la consommation immédiate. Toutefois, le sentiment général est, de la part de tous les artistes qui ont eu à ce jour l'occasion d'aborder ce concept et ce système, que l'apprentissage est véritablement possible, que les concepts de base en très petit nombre et compréhensibles de manière complète encouragent à s'y engager.

Souvent, des réflexes issus des approches plus classiques relevant par exemple des systèmes fondés sur les modèles de signaux, prennent le dessus et induisent le désir d'une traduction directe. Il est important à ce moment précis de montrer que les intentions musicales sous-jacentes peuvent, ce qui est très généralement le cas, trouver un pendant beaucoup plus pertinent dans le formalisme modèle physique. Le « penser physique » que nous avons évoqué tout au long de ce document, prend ici tout son sens. Dans tous les cas où cette situation s'est présentée, les artistes concernés ont très rapidement reconnu, avec un certain enthousiasme, avoir franchi une étape dans leurs propres représentations et leurs modes opératoires.

D'une manière générale, si le cheminement de pensée qui consiste à tenter de traduire systématiquement un phénomène, une figure, un développement sonore prédéfinis et dont la représentation mentale est déjà relativement précise en termes des processus physiques dynamiques (objets physiques simples, objets en interaction) qui les engendre ou engendre certains de leurs attributs, est véritablement difficile au début, nous avons constaté qu'il pouvait s'apprendre et se développer d'une manière assez surprenante. Par ailleurs, parallèlement à ce mode de création déterminé par la préconception d'une idée ou d'un but à atteindre, la propriété des objets physiques virtuels d'exhiber des comportements inattendus (mais toujours colorés de *plausibilité* et de cohérence physique) est très stimulante dans une phase d'exploration créatrice.

Si ces potentialités étaient présentes dans le formalisme CORDIS-ANIMA, ce n'est qu'à travers une interface qui les met en perspective que l'artiste peut véritablement les inscrire dans ses modes opératoires.

Au delà de cet aspect, l'interface permet de soumettre à l'expérience des principes qui ont été imaginés de longue date comme bien sûr la synthèse du matériau sonore à l'aide des modèles physiques, mais aussi et peut-être surtout, celui de la création musicale par « composition de modèles physiques ». De telles expériences ont en effet été réalisées pour la première fois par des artistes collaborateurs du projet, il y a peu de temps, à l'aide de l'une des dernières versions du logiciel (proche de la version actuelle). Ces expériences ont débouché sur la présentation de plusieurs œuvres musicales dont l'originalité et la pertinence ont été soulignées.

Sans doute le point d'aboutissement de nos travaux à travers cet outil apparaît-il en premier lieu sous un angle technologique : nous avons développé un outil qui peut prétendre apporter quelques nouvelles latitudes au processus de création musicale informatique. Mais nous croyons aussi qu'un outil, tout matériel qu'il soit, est aussi une théorie, ou en tout cas qu'il peut la porter, la représenter, et permettre de l'expérimenter.

Annexes

Annexe A : Modelisation physique, Son, Musique _____ **354**

Annexe B : Etude d'un objet CORDIS-ANIMA bidimensionnel **385**

Annexe C : « Simuler pour connaître ; Connaître pour simuler » (Claude Cadoz) _____ **393**

Annexe D : Quelques notes sur l'évaluation _____ **433**

Annexe E : Plages couleur _____ **443**

Annexe A
Modélisation physique, Son, Musique

Le *modèle physique* est de plus en plus utilisé dans le contexte de la création musicale. Nos travaux participent de ce mouvement : avec GENESIS, il s'agit de permettre une démarche de création complète autour du paradigme CORDIS-ANIMA. Dans cette première annexe, nous nous proposons de placer quelques jalons autour de la notion de *modèle physique* et des différentes techniques qui s'en réclament.

Le premier paragraphe précise quelques définitions puis propose une synthèse des attentes nombreuses et diverses que suscite auprès des chercheurs et des musiciens le développement des modèles physiques.

Le second paragraphe introduit deux séries de *critère d'évaluation comparée* des techniques de synthèse sonore. La première, inspirée de l'approche de Jaffe [Jaffe.95], concerne aussi bien les modèles physiques que les techniques de synthèse plus traditionnelles. La seconde, devenue nécessaire compte tenu des développements des techniques de modélisation physique, est entièrement consacrée à ces dernières.

Dans le troisième paragraphe, nous présentons succinctement les diverses techniques de modélisation aujourd'hui pratiquées dans le contexte de l'Informatique Musicale. Chacune d'entre elle est brièvement étudiée sous l'angle des critères précédemment proposés. Enfin, nous positionnons le formalisme CORDIS-ANIMA en regard des autres approches du *modèle physique*.

1 - Le *modèle physique*, une notion porteuse d'espérances _____ 355

1.1 Quelques définitions 355
1.2 Modèles de signaux / modèles physiques 356
1.3 Le modèle physique en question 358
1.4 Son d'instruments traditionnels ; mimétisme 359
1.5 Les modèles physiques, "générateurs de réalité musicale" 360
1.6 Modèles physiques, causalité acoustique et objectif de "plausibilité" 361
1.7 Modèle physique, comportement dynamique, jeu ; accident et phrasé 362

2 - Des critères pour une évaluation des techniques de synthèse et de modélisation physique pour la musique _____ 364

2.1 Des critères pour une évaluation des techniques de synthèse 365
2.2 Evaluation globale des modèles physiques en regard des critères 371
2.3 Des critères spécifiques aux modèles physiques 372

3 - Principales techniques de modélisation physique pour la musique _____ 375

3.1 Analyse numérique, éléments finis, différences finies... Les outils de l'acousticien 376
3.2 L' "approche classique" des travaux de Ruiz et Hiller 376
3.3 Les guides d'ondes 377
3.4 L'approche source / filtre en interaction, techniques numériques non linéaires 379
3.5 La synthèse modale 380
3.6 L'approche particulaire□CORDIS-ANIMA 383

1. Le *modèle physique*, une notion porteuse d'espérances

1.1. Quelques définitions

Les premiers sons numériques de synthèse sont apparus en 1957 suite à la mise au point du convertisseur numérique/analogique par Max Matthews aux Bell Laboratories. Les chercheurs de cette époque pionnière ont pu considérer que *tout* l'univers sonore était désormais accessible, que tous les sons étaient désormais possibles. Cependant, sans même parler des problèmes liés à la transduction du signal sonore (rayonnement, diffusion, effets de localisation...), cet infini espéré s'avère des plus difficiles à atteindre : *«* Restaient alors à définir des stratégies pour partir à la conquête de ces territoires nouveaux et inexplorés, faute de quoi la carte risquait de se confondre avec le territoire » [Barrière.90]. Ces stratégies sont musicales mais aussi techniques et scientifiques : si tous les signaux sonores numériques sont potentiellement possibles, les chemins permettant d'atteindre les caractéristiques attendues pour un son exploité musicalement sont très vite complexes.

La problématique générale de la synthèse sonore pourrait se formuler ainsi :

Etant donné la finitude de l'espace des signaux numériques calculables,
étant donné que tous les sons numériques peuvent potentiellement être générés par l'ordinateur,
comment établir des projections opératoires d'un point de vue perceptif dans l'espace des signaux numériques sonore ?

Autour de cette problématique s'organisent plusieurs recherches non moins importantes relevant d'autres disciplines[1]. Mais, après 45 ans d'histoire de la synthèse numérique, la question des *techniques de synthèse* demeure centrale.

Une technique de synthèse numérique « est un ensemble de procédés de calculs et de formules mathématiques, qui calculent chacune des valeurs des échantillons » [DePoli.83]. C'est, en conséquence, une classe d'algorithmes permettant la génération du signal. A ce titre, CORDIS-ANIMA peut être considéré comme une technique de synthèse.

Nous appellerons *modèle* une instance d'une telle classe d'algorithmes (figure 1). Un modèle laisse libres plusieurs *paramètres* que peut choisir l'utilisateur. Certains de ces paramètres sont fixés avant l'exécution de l'algorithme ; nous appellerons alors *instrument numérique* (comme le propose De Poli, [DePoli.83]) un modèle avec ses paramètres, c'est à dire un algorithme susceptible d'être mis en calcul. Enfin, certains paramètres du modèle - dont le type peut être imposé par la technique - sont variables pendant la durée du calcul. Nous appellerons alors *flux dynamique* ou *flux paramétrique* l'ensemble de leurs valeurs au cours du temps.

[1] Philippe Depalle et Xavier Rodet écrivent : "Selon l'acception courante, la synthèse des sons est l'activité qui consiste à créer de nouveaux sons à partir de dispositifs électroniques ou informatiques. Mais, à la réflexion, cette définition ne recouvre qu'une partie de la réalité. La synthèse des sons est plutôt une discipline qui trouve son sens dans la réunion et l'articulation de quatre champs d'étude consécutifs : la conception de modèles de synthèse, l'analyse des sons, la réalisation de dispositifs de synthèse et le développement de stratégies de contrôle de ces dispositifs" [Depalle&Rodet.93].

Une technique de synthèse

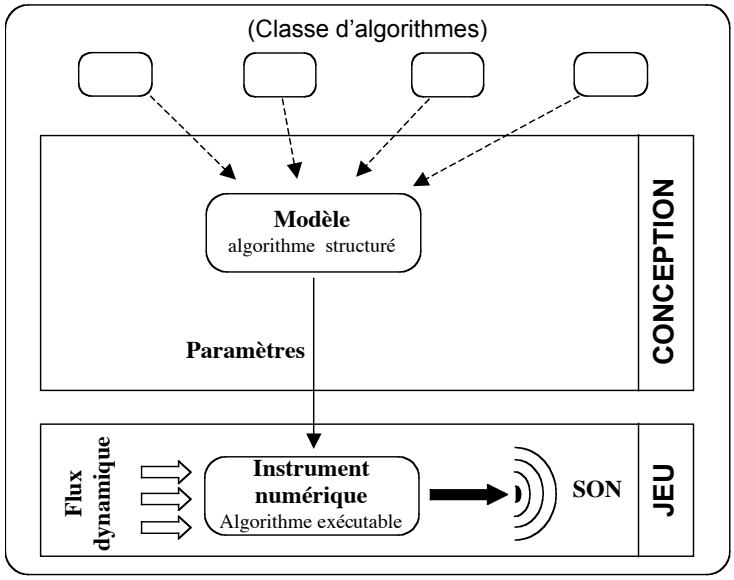


Figure 1 : définitions des techniques de synthèse
Modèle, instrument, paramètres, flux dynamique

Ces définitions font apparaître l'existence de trois niveaux de travail autour des techniques de synthèse. Le premier niveau est structurel. Il consiste à déterminer la structure de l'algorithme du modèle. Le second est paramétrique. Il permet de rendre calculable l'algorithme. Le dernier est un flux dynamique continu ou discret, uni ou bi directionnel, qui modifie le modèle pendant la simulation, que celle-ci soit temps réel ou temps différé. Il induit la possibilité de jouer sur l'instrument synthétique.

Enfin, précisons dès maintenant quatre caractérisations possibles des paramètres. Les *paramètres acoustiques*[1] seront ceux que l'on peut mesurer sur le signal sonore : fréquence, amplitude, contenu spectral... Les *paramètres perceptifs*, parfois appelés psychoacoustiques, seront ceux que définit la psychoacoustique quand elle se réfère à des dimensions (supposées indépendantes) de la perception : hauteur, sonie, timbre par exemple. Enfin, nous appellerons *paramètre signal* tout paramètre d'une technique de synthèse par modèles de signaux et de la même façon *paramètre physique* ceux des techniques par modèles physiques.

1.2. Modèles de signaux / modèles physiques

Aujourd'hui, la plupart des modèles de synthèse conduisent à des *modèles de signaux.* Dans cette approche, le signal numérique est pris comme objet ; c'est lui qui est façonné, généré par étapes successives.

Les modèles de signaux permettent un contrôle précis des paramètres acoustiques du son, une maîtrise complète de la structure des sons. Complexifiés et diversifiés au cours des ans, implémentés dans des environnements de plus en plus élaborés[2], ils ont permis la pratique de la synthèse avec un haut niveau d'exigence musicale et en ont fait une activité

[1] Ils sont parfois appelés paramètres physiques (du son) - nous réservons ces termes à une autre signification.

[2] Plusieurs logiciels (programmes MUSIC et descendants, voir par exemple [Roads.80]) et dispositifs matériels (synthétiseurs numériques) permettent de pratiquer les différentes techniques en modèles de signaux dans des environnements unifiés.

importante dans l'univers des musiciens. Leur diffusion s'est accompagnée du développement d'un vocabulaire de description des sons, désormais couramment utilisé par les musiciens – et pas seulement ceux qui pratiquent la synthèse : Attaque, *decay*, fréquence, partiel, banc de filtres, etc.

Aujourd'hui, l'approche *signal* du son musical est profondément enracinée – à tel point qu'il est difficile à un compositeur de s'adapter à une nouvelle approche sans y faire constamment référence. C'est pourtant là ce que proposent les *modèles physiques*, en déplaçant l'attention du signal (et de ses paramètres) vers la *cause* qui le produit.

Les instruments traditionnels sont des dispositifs que des siècles d'élaboration et d'améliorations empiriques ont façonnés et dont l'étude est d'un grand intérêt pour les acousticiens[1]. Il n'est guère étonnant que la simulation de modèles physiques, en informatique musicale, ait été utilisée par les acousticiens dans l'objectif de valider des modèles d'instruments. Il s'agissait de "simuler pour connaître" (selon l'expression de Claude Cadoz, [Cadoz.90a][2]), c'est-à-dire pour progresser dans la connaissance des principes physiques fondamentaux mis en œuvre dans les instruments.

Dès les années 67-70, dans sa thèse sous la direction de Lejaren Hiller[3], Pierre Ruiz conçoit un modèle numérique de corde dans un but non plus d'étude mais *de « synthèse de son »* [Hiller.71]. Mais les *modèles physiques* ont véritablement débuté leur parcours dans la musique avec d'une part les travaux de l'ACROE (1979 marque les débuts d'une réflexion sur la Simulation Multisensorielle Interactive d'Objets Physiques et son impact dans le processus de création, [Cadoz.79]), d'autre part ceux de Kevin Karplus et Alex Strong (avec la conception en 1979 de l'algorithme Karplus-Strong, célèbre par l'élégante solution apportée au problème du temps de calcul pour un son réaliste, [Karplus.83])[4].

Il est difficile de dégager une définition des *modèles physiques* qui serait commune à l'ensemble des chercheurs qui se réclament de l'approche[5]. Nous retiendrons ici la proposition introduite lors du colloque international sur les Modèles Physiques de Grenoble en 1990 [ColModPhys.90] : une synthèse par modèle physique consiste à simuler les causes plutôt que synthétiser les effets.

L'apparition des Modèles Physiques pour la synthèse correspond ainsi à une évolution historique, naturelle en ce qu'elle va du plus immédiat au plus profond. Les premières techniques de synthèse visaient à dégager les algorithmes minimaux permettant la génération d'un signal acoustique. Dans une seconde étape, l'objectif fut de simuler un phénomène sonore en s'appuyant sur toute la connaissance acquise sur ce dernier par des années d'expérimentation et d'analyse. Avec les techniques par modèles physiques, il ne s'agit plus de représenter et simuler le phénomène, mais la cause sous-jacente qui le produit et l'explique.

---

[1] Voir par exemple [Causse.90], [McIntyre&Woodhouse.79], [Causse.93].

[2] Ce texte est reproduit *in-extenso* dans l'annexe C.

[3] L'introduction du modèle physique pour la synthèse du son musical s'est donc faite sous la direction du pionnier de la composition assistée par ordinateur.

[4] Dès les années 80, l'algorithme permet la synthèse simultanée en temps-réel des sons de 16 cordes pincées. Il faut noter, toutefois, que Karplus et Strong ne se réclamaient pas d'une démarche de modélisation physique : il s'agissait encore de *synthétiser* des sons de cordes, et non de *modéliser* ces dernières. A notre connaissance, ce n'est qu'en 1983 que Julius O. Smith montre qu'une interprétation « physique » de l'algorithme Karplus-Strong est possible, autour de la notion de *propagation*. Cette ré-interprétation le conduira à introduire le principe des guides d'ondes, que nous présentons plus loin.

[5] L'annexe C reproduit *in-extenso* le texte *Simuler pour connaître, Connaître pour simuler* de Claude Cadoz, qui offre quelques éléments de réponse en la matière.

1.3. Le modèle physique en question

La validité d'une telle démarche a mis du temps à s'imposer. Il est vrai que simuler un objet physique pour en *extraire* un son ne semble pas une voie économe.

Les modèles physiques sont par nature plus coûteux en temps de calcul que les modèles de signaux. C'est encore aujourd'hui l'un des obstacles majeurs à leur mise en œuvre. Le son synthétisé, objectif du processus, n'est qu'une facette des phénomènes simulés – et on peut se demander si la simulation du reste est pertinente.

Mais l'approche par modèle de signaux ne se justifie pas seulement par l'efficacité des algorithmes correspndants.

> « La musique est faite pour être entendue, et a priori les modèles physiques ne sont pas spécialement pertinents parce que ce qui compte n'est pas le processus de production mais la perception, l'effet sur les sens, la sanction phénoménologique » [Risset 90].

Depuis les années 60, la connaissance des propriétés du signal sonore musical et des rapports complexes entre paramètres acoustiques (mesurables) du signal a considérablement progressé. Parallèlement, les techniques de synthèse signal permettent la génération de sons de plus en plus complexes et une prédiction toujours meilleure de leurs effets sur la perception. Dans ces conditions, n'est-ce pas s'éloigner du musical que de considérer des propriétés de la *matière* alors qu'on sait engendrer et qualifier les effets, qui semblent bien être la raison pour laquelle on recourt à la synthèse ?

Cette question semble d'autant plus légitime que ces *modèles physiques* paraissent particulièrement difficiles à mettre en œuvre : « La construction d'un modèle physique est en général une entreprise longue et délicate » et « la complexité du modèle physique peut être un frein à son développement, alors que la même somme d'efforts consacrée à un modèle du signal peut permettre de le porter à un haut degré de perfectionnement » [Rodet&al.90]. Les paramètres sont souvent éloignés des paramètres perceptifs qu'on a eu tant de mal à caractériser : « Contrairement aux modes de synthèse sonore où l'on met en jeu directement les paramètres sonores, la variation d'une valeur d'un paramètre physique pourra avoir sur le son des conséquences multiples agissant simultanément sur la hauteur, la durée et le timbre. Le réglage en est donc assez délicat » [Loizillon.96].

Enfin, on peut s'interroger, avec Risset, sur les buts de la synthèse sonore :

> « Avec la synthèse, on cherche à dépasser les limitations des systèmes mécaniques. Aussi, à première vue, utiliser l'ordinateur pour mettre en œuvre des modèles physiques peut sembler entrer dans l'avenir à reculons, comme disait Mac Luhan, puisque c'est chercher à reproduire ce qui existe déjà dans le monde acoustique » [Risset.90].

Ainsi dans les années 80, le modèle physique parait pour beaucoup trop ancré dans un réel qu'on souhaite dépasser.

Malgré ces importantes interrogations, les modèles physiques ont continué à se développer sous l'impulsion commune des acousticiens, des informaticiens, des spécialistes en traitement du signal et à la demande des musiciens[1]. Tous entretiennent de fait des attentes fortes à l'égard de la modélisation physique.

[1] La place qui leur est accordée dans les publications scientifiques dans le domaine de l'Informatique Musicale n'a cessé d'augmenter. On peut remarquer que la majorité des innovations récentes en matière de synthèse se réclament du *modèle physique*. Parallèlement, les compositeurs utilisent de plus en plus ces nouvelles techniques, et l'industrie s'en est emparée pour concevoir certains des synthétiseurs récents (Korh, Yamaha, Roland ou Technics par exemple).

1.4. Son d'instruments traditionnels ; mimétisme

La première de ces attentes s'oppose justement à ce que nous venons de dire. Elle réside dans l'aptitude des modèles physiques à imiter les sons des instruments traditionnels.

Le son des instruments traditionnels constitue une référence culturelle commune. En conséquence, la simulation des phénomènes sonores qu'ils engendrent est depuis toujours un objectif de la synthèse, et ce, quelle que soit la technique considérée. Jean-Baptiste Barrière voit en cela une démarche naturelle, conséquence de ce qu'il nomme l'hypothèse de continuité : la synthèse numérique des sons offre tout son intérêt non pas quand elle transgresse la tradition auditive mais quand elle se situe en tant qu' « actualisation d'un potentiel, transformation et prolongation » d'une culture du sonore façonnée par des siècles d'histoire [Barrière.90]. Par ailleurs, il faut bien remarquer que l'utilisation la plus répandue des algorithmes de synthèse sonore, dans la musique populaire notamment, est la mise à disposition d'ersatz d'instruments traditionnels.

Les échantillonneurs (*sampling-synthesis)* parviennent bien à réaliser une telle illusion et c'est en cela qu'ils rencontrent un important succès. Les modèles physiques sont cependant une alternative intéressante, car plus générative et plus riche. Smith écrit ainsi :

> « Nous nous approchons du moment où de nombreux instruments virtuels seront interchangeables avec leur vis-à-vis réel. Déjà, les échantillonneurs permettent une complète interchangeabilité dans le cas d'une note unique – la note qui a été échantillonnée. Les modèles physiques, cependant, commencent à offrir une parité pour une variété d'expressions plus importante, et requièrent nettement moins de mémoire (mais plus de puissance de calcul, il est vrai) pour atteindre cette qualité. (…) En fait, les modèles physiques peuvent être vus comme une forme 'd'échantillonnage structuré' dans lequel des paramètres physiques plus profond sont 'échantillonnés' à la place des fluctuations de la pression acoustique enregistrée dans les échantillonneurs traditionnels » [Smith.96][1].

Une telle *visée mimétique* constitue pour beaucoup l'intérêt majeur des modèles physiques – position confortée en ce qu'elle rencontre l'attente de nombreux utilisateurs, et par la même qu'il s'appuie sur un potentiel commercial considérable. Mais est-ce fondamental ? Est-ce tirer le meilleur parti de la synthèse ? Est-ce la base d'une possible évolution du processus de création et d'un possible renouveau de la Musique ? A l'évidence, non, comme le rappelle Caussé :

> « Utiliser ces modèles de synthèse lors d'applications musicales pour imiter les instruments "réels" n'a certainement pas grand intérêt et en parodiant G. Hegel on peut dire que développer de tels instruments est probablement 'un travail superflu... un jeu présomptueux. L'imitation ne pouvant produire que des Chefs-d'œuvre de technique, jamais des œuvres d'art' (Hegel.1821, l'Esthétique) » [Caussé.90].

Le mimétisme, heureusement, n'est pas le seul intérêt des *modèles physiques.*

---

[1] Traduction libre de l'auteur – *"Historically, we are approaching parity between real and virtual acoustic instruments in the context of recorded music playback. That is, we are approaching the time when many virtual instruments can be considered interchangeable with their real-world counterparts for recording purpose. Already, sampling synthesis gives us full interchangeability for the case of a single note - the note that was sampled. Model based techniques, however, are beginning to provide parity over a wider variety of performance expression, and they require far less memory (though more computational power) to achieve this quality. (...) In fact, physical models can be seen as a form of 'structured sampling synthesis' in which deeper physical parameters are sampled in place of the simple air pressure fluctuations recorded in traditional sampling synthesis".*

1.5. Les modèles physiques, "générateurs de réalité musicale"[1]

Maintenant que nous avons des modèles physiques capables d'imiter le son des instruments, rien n'interdit de les pervertir, de les distordre. Borin & al considèrent alors que les algorithmes par modèles physiques deviennent essentiellement des *générateurs de réalité musicale* qu'on peut considérer en eux-mêmes sans plus les juger à l'aune d'une quelconque réalité :

> « De notre point de vue, la synthèse par modèles physiques – devenue une sorte de 'générateur de réalité musicale' en elle-même – permet de prendre son inspiration dans le monde réel afin de décliner notre interprétation de ce dernier sans que nous soyons forcés de limiter nos expérimentations aux équations physiques usuelles. […] Il pourrait être particulièrement intéressant de créer des structures qui n'ont pas d'interprétation physique et ne respectent que la contrainte d'être stables et passives. Dans ce cas, la réalité physique est considérée comme une source d'inspiration et non comme une référence pour un jugement qualitatif des sonorités produites » [Borin&al.92a][2].

Partant d'un modèle physique (d'un instrument) ou d'une classe de modèles, il est possible de construire un autre modèle sans référence réelle - dans la suite nous continuerons à qualifier de "physique" ce second modèle, car il met en œuvre des "blocs" et des procédés inspirés de la modélisation d'un réel (nous y reviendrons). Comment, alors, caractériser le potentiel sonore des modèles physiques ?

Pour Borin & al, « les propriétés des sonorités produites peuvent être identifiées à la structure de la classe d'algorithmes qui les ont générées » [Borin&al.92a][3] et la structure des algorithmes des modèles physiques a ceci de particulier qu'elle est constituée de plusieurs blocs en interaction[4]. Selon les mêmes auteurs, cela garantit la génération de sons « homogènes » et « riches », et ce même s'ils ne sont plus évocateurs d'une réalité physique. *L'homogénéité* du son ne nous semble cependant pas décrire suffisamment le caractère des sons espérés avec les modèles physiques ; nous introduirons la notion de *plausibilité* des sons.

---

[1] L'expression est extraite de [Borin&al.92].

[2] Traduction libre de l'auteur - *"From our point of view, synthesis by physical models - being a sort of musical reality generator on its own - makes it possible to take inspiration from the real world in order to derive our interpretation of it without forcing us to limit experimentation to the usual physical equations. (...) It might be particularly interesting to create structures that are not required to have any physical interpretation and have the only constraint of being stable and passive. In this case, physical reality is taken only as a source of inspiration and is not used as a reference for a qualitative judgement of the sonority produced"*.

[3] Traduction libre de l'auteur - *"the properties of the produced sonorities can be identified with the structure of the class of algorithms to which it belongs"*

[4] Borin & al distinguent les structures de génération directe, dans lesquelles le son résulte de la sommation de blocs indépendants (synthèse additive par exemple), les structures en cascade de blocs, dans lesquels la sortie d'un bloc est l'entrée du suivant (synthèse FM par exemple) et les structures en interaction. Les modèles de signaux mettent en œuvre les deux premiers types de structures, les modèles physiques la troisième [Borin&al.92].

1.6. Modèles physiques, causalité acoustique et objectif de "plausibilité"

« C'est nier l'évidence que de croire que la musique puisse dispenser l'oreille de sa fonction la plus essentielle : celle de renseigner l'homme sur les évènements qui surviennent » [Schaeffer.66].

Lors du colloque Modèles Physiques de Grenoble en 1990 [ColModPhys.90], Jean-Claude Risset a exposé certaines des spécificités de la perception qu'il est intéressant de considérer lorsqu'on aborde les *modèles physiques* :

> « En fait, notre perception n'est pas neutre. Elle a été modelée par l'évolution des espèces, la phylogenèse, ou de l'individu, l'ontogenèse, qui l'ont adaptée à un monde physique réel. (...) La psychophysique traditionnelle, depuis Weber et Fechner, s'efforce de dégager des échelles sensorielles indiquant de quelle façon les sens évaluent les paramètres physiques [*acoustiques*]. Mais Gibson a fait remarquer que là n'est pas la fonction sensorielle primordiale : pour lui, nos sens ont évolué non pas pour mesurer au mieux le monde physique mais pour extraire à partir des données sensorielles des informations sur ce monde physique pouvant aider à la survie. (...) En fait, l'oreille a sans doute pour fonction originaire, outre la mise en alerte, d'effectuer une véritable enquête sur l'origine du son : d'où vient-il (distance et azimut), et comment a-t-il été produit. » [Risset.90].

Ce caractère de l'audition joue un rôle déterminant dans l'organisation de la perception. Observé relativement aux sons d'origine acoustique, sa généralisation permet d'inférer des propriétés souhaitables pour les sons de synthèse :

> « En particulier, les timbres synthétiques ont plus de relief, ont une identité beaucoup plus marquée, plus robuste, lorsqu'on peut imaginer de quelle manière ils auraient pu être produits physiquement ; et l'oreille est à l'affût d'incidents, d'idiosyncrasies du son qui l'aident à remonter à son origine » [Risset.90].

On pourrait donc dire qu'un certain « réalisme » peut être recherché pour les sons de synthèses. *Réalisme* renvoie cependant à ce réel dont justement on veut par la synthèse s'affranchir. Pearson qualifie ses attentes vis à vis des sons de synthèse, qui l'ont conduit à aborder les modèles physiques, par les trois adjectifs « organique, complexe et cohérent » [TAO]. Nous préférons quant à nous introduire la notion de *plausibilité physique*. Un son peut ne correspondre à aucune source acoustique, il peut n'en évoquer aucune, tout en étant *physiquement plausible.* Il nous semble que ce qui importe pour qu'un son soit valide au regard du caractère de l'audition pointé par Risset, ce n'est pas tant qu'il permette d'inférer son origine mais plutôt qu'il renferme un ensemble de caractères intimes, de liens dynamiques subtils entre paramètres acoustiques qui le rendent *physiquement plausible*.

Un musicien est libre de vouloir éviter tout encrage dans l'univers acoustique lorsqu'il fait appel à la synthèse sonore et peut-être ainsi interroger voir choquer l'oreille. Cependant, dès lors qu'il fera sien « l'objectif de plausibilité », les modèles physiques présenteront pour lui un intérêt évident :

> "[la causalité,] c'est ce qui fait l'intérêt des modèles physiques. (...). En modifiant les paramètres du modèle physique, on a des chances d'agir sur des aspects du son qui sont robustes, prégnants, pertinents pour l'audition" [Risset.90].

Toute technique de synthèse peut se prêter à la recherche d'une plausibilité. Les modèles physiques cependant la portent potentiellement dans leur structure même, là où il faudra un jeu savant de paramètres (dynamiques ou non) pour l'obtenir avec un modèle de signal. Or, comme le remarquent Borin & al, il est sans doute souhaitable que la complexité du son – ici sa *plausibilité* – soit une propriété native du mécanisme de production plutôt qu'une propriété construite artificiellement par l'utilisateur. Les modèles physiques offrent *par nature* une permanence de la plausibilité lorsqu'on "pervertit" un modèle pour en créer un autre, une robustesse face aux variations.

1.7. Modèle physique, comportement dynamique, jeu ; accident et phrasé

Nous envisageons plus explicitement ici, dans la continuité de ce qui précède, le comportement dynamique des modèles physiques en situation de jeu (temps réel ou temps différé).

Les modèles physiques induisent une partition nette entre paramètres statiques - responsables de la permanence de l'instrument synthétique - et flux paramétrique. Rodet & al observent que :

> « dans le cas du modèle physique, certains paramètres peuvent être un peu difficiles à concevoir par l'utilisateur, tel le module d'Young d'une corde. [...] D'autres paramètres en revanche ont une pertinence immédiate dans le cadre du geste instrumental, telle la force ou la vitesse d'un marteau » [Rodet&al.90].

Les accès dynamiques aux modèles physiques autorisent ainsi une représentation particulièrement naturelle des actions que tous les jours nous avons vis-à-vis des objets réels, qu'ils permettent une métaphore du geste instrumental. C'est, comme l'explique Vinet, une caractéristique essentielle :

> « [le *modèle physique*] c'est une nouvelle technique de synthèse qui se propose à nous, et en cela nous apporte une nouvelle richesse de sons réalisables. Cela est le premier côté positif : la richesse des sons qu'on peut obtenir par les modèles physiques et qu'on ne peut pas atteindre nécessairement avec des modèles de synthèse plus traditionnels. Deuxième aspect : le corps a sa propre pensée, le corps au sens de motricité. (...) L'intérêt du modèle physique dans une situation de musique qui n'est pas complètement une situation d'écriture, et qui inclut une situation expérimentale, c'est de restituer des aspects qui sont plus directement 'préhensibles' par le corps » (Hugues Vinet, [TableRonde.90b]).

C'est en soi intéressant, mais ça l'est d'autant plus que les flux dynamiques influent sur plusieurs paramètres perceptifs en même temps[1]. C'est cela qui précisément permet de conserver l'illusion d'une permanence de la cause. Nous avons déjà cité Loizillon selon qui les paramètres des modèles physiques sont délicats à manipuler car ils peuvent influer simultanément sur plusieurs paramètres acoustiques et perceptifs du son, alors que d'autres techniques offrent un contrôle indépendant pour chacun d'entre eux. Cette propriété peut être envisagée suivant la position inverse, en remarquant que la corrélation des dimensions psychoacoustiques induite par les paramètres des modèles est, au contraire, gage de cohérence, comme le reconnait plus loin Loizillon :

[1] A priori rien n'interdit que le module d'Young, par exemple, soit contrôlable dynamiquement, et on peut espérer que le naturel de la corrélation des dimensions de la perception que nous évoquions plus haut soit conservée.

> « Les éléments qui, dans la synthèse par modèle de signal sont explicitement spécifiés (fréquence, durée, profil dynamique) deviennent en synthèse par modèles physiques des conséquences de la spécification de la forme et de la matière des objets et de modes d'interactions entre ceux-ci. Nous sommes, en ce sens, réellement placés dans certaines des conditions instrumentales ; celles de l'exploration des capacités sonores d'un dispositif mécanique » [Loizillon.96].

D'Alessandro et Beautemps font une remarque similaire mais plus explicite à propos de cette corrélation des paramètres perceptifs. Observant que les variations des paramètres acoustiques sont contraintes par l'essence même du modèle ils soulignent qu'on « peut espérer que les évolutions des paramètres acoustiques obtenues par un tel synthétiseur [par modèle physiques] soient automatiquement des évolutions plausibles en faisant l'économie de les spécifier explicitement » [D'Alessandro&Beautemps.90].

Ainsi, les modèles physiques déplacent l'origine des subtiles variations dynamiques responsables d'une partie de la « vie » du son du flux paramétrique vers le modèle lui-même. Ils permettent la génération automatique de formes sonores à court terme (on pourrait dire "micro-structurelles") en réponse à un "*input*" simple - pousser, tirer, appuyer, etc.

De là, il n'y a qu'un pas à faire pour en venir à la notion de *phrasé*. Face à un instrument synthétique temps réel ou temps différé, on comprend que le phrasé possible est lié aux accès dynamiques mis à disposition du musicien et à la pertinence de ces accès. On comprend aussi qu'elle dépend de la capacité du modèle à générer des micro-variations subtiles et prégnantes, ainsi que des « accidents » dans le son, des petites variations peu prévisibles. De nombreux auteurs s'accordent alors à penser que les modèles physiques permettent un phrasé plus immédiat et convainquant que les modèles de signaux.

Constatons à ce stade que l'on a réalisé une boucle sur l'approche originelle du geste instrumental par l'ACROE. Dans la partie I, nous avons montré comment le programme de recherche du laboratoire, partant de la relation instrumentale et du désir de réhabiliter la sensibilité gestuelle dans un contexte informatique, a introduit les périphériques spécifiques à retour d'effort puis le système CORDIS-ANIMA de modélisation physique. À l'inverse, ce que constatent aujourd'hui les personnes impliquées dans la synthèse sonore par modèles physiques, c'est que ceux-ci sont *effectivement* pertinents pour un renouveau du potentiel expressif des sons synthétiques, et que, par la métaphore naturelle des flux dynamiques qu'ils permettent, ils se prêtent particulièrement bien à un contrôle gestuel.

Nous fermons incidemment une seconde boucle, celle-ci avec le paragraphe précédent. La corrélation entre paramètres perceptifs induite par les modèles physiques peut être envisagée comme une globalisation de la perception. Les modèles de signaux permettent une indépendance entre paramètres acoustiques et par là même ont accompagné l'essor de la psychoacoustique traditionnelle et la définition d'une base de paramètres perceptifs. Les modèles physiques s'adressent à la perception de manière globale et ils le font de manière pertinente en regard de la notion de causalité. Là où les modèles de signaux codent des dimensions de la perception considérées comme indépendantes, les modèles physiques représenteraient la perception dans son ensemble. Dès lors, peut-être vont-ils à leur tour « permettre de mieux comprendre les particularités de l'audition » [Risset.90], et peut-être verrons-nous émerger, en suivant le point de vue de Gibson, une « psychoacoustique écologique » qui s'appuierait sur une simulation physique du perceptible.

2. Des critères pour une évaluation des techniques de synthèse et de modélisation physique pour la musique

En quelque 40 ans, de très nombreuses techniques de synthèse ont été expérimentées - ainsi qu'un nombre encore plus important de modèles. Chacune d'entre elles, qu'elle repose sur une approche *signal* ou *physique*, favorise certaines variations dynamiques et s'avèrent plus ou moins accessible et performante suivant le contexte dans lequel elle est utilisée :

> *«* Si l'on déploie suffisamment d'énergie et d'imagination, on peut arriver à une synthèse de bonne qualité avec chaque technique, mais les chances de succès ne sont pas les mêmes au départ, il existe des points forts et faibles constitutionnels à chaque technique *»* [Barrière.90].

Il est donc nécessaire de disposer de quelques jalons face à la diversité des techniques de synthèse. Plutôt que de présenter d'emblée des considérations techniques, nous nous placerons d'abord dans la situation d'un utilisateur qui aurait à choisir entre différentes approches et proposerons des *critères pour une évaluation des techniques de synthèse.* Ce faisant, nous visons également à offrir une présentation synthétique des différents aspects, enjeux, questions adressées par la synthèse des sons.

Une telle approche est rare dans la littérature - probablement parce qu'un consensus est difficile à dégager. David A. Jaffe est le seul à notre connaissance à l'avoir adoptée explicitement [Jaffe.95]. Sur la base des 10 critères qu'il propose, Jaffe conclut à l'intérêt de la modélisation physique pour la synthèse sonore. Ces critères, cependant, s'avèrent insuffisants dans le contexte de l'extension des concepts de la synthèse suscité par le développement et la diversification des modèles physiques. Aussi, nous avons cherché une base étendue de critères qui permettent une évaluation comparée des diverses techniques de modélisation physique aujourd'hui pratiquées en Informatique Musicale.

Dans la suite, nous proposons successivement :

- 13 critères généraux pouvant s'appliquer aux *techniques de synthèse,* qu'elles soient en « modèles de signaux » ou en « modèles physique ». Les critères directement inspirés de ceux que propose Jaffe sont repérés par un astérisque « * ».
- Une évaluation globale du paradigme de la modélisation physique en regard de ces 13 critères.
- 6 critères complémentaires propres aux techniques de modélisation physique qui permettent de mieux comprendre leurs différences essentielles.

Nous nous sommes efforcés de rechercher une base de critères *indépendants*. Mais, s'agissant de critères, l'orthogonalité n'est pas chose facile à obtenir. Certains restent donc plus ou moins corrélés. Par ailleurs, la satisfaction de certains critères n'est pas toujours souhaitable : selon les attentes de l'utilisateur, il peut sembler préférable de les mettre en avant ou au contraire de les minorer.

Aussi, ces 13 critères ne doivent-ils pas être entendus comme des outils de *classement*. Ils constituent plutôt une grille partielle de comparaison (ou d'évaluation comparée) des potentiels des diverses techniques ou encore une présentation synthétique des multiples propriétés que chercheurs et musiciens peuvent attendre d'une technique de synthèse.

Critères relatifs aux sons productibles

C1 : La technique permet-elle la synthèse de sons « musicalement riches » ?

Il s'agit là d'un souhait légitime, mais bien difficile à sanctionner de façon objective tant la notion de "richesse musicale" peut prêter à tergiversation.

C'est que tout son de synthèse peut être intéressant dans un contexte musical donné, sa « valeur », n'étant définissable que partiellement par ses propriétés intrinsèques. Le contexte dans lequel il s'insère et, par-dessus cela, le choix esthétique du compositeur jouent un rôle inaliénable, chaque musicien étant seul habilité, en dernière analyse, à juger la richesse musicale, par la « sanction phénoménologique » [Risset.90]. Il existerait, alors, autant de façons d'envisager la « richesse musicale » que de compositeurs, de musiciens, voire d'auditeurs – et ce critère devient caduc.

Une manière « objective » d'aborder ce critère pourrait consister à l'appliquer à une observation du signal et de l'associer à la complexité de sa structure dynamique. Ainsi, pour Borin & al : « un 'bon' son est caractérisé par la complexité de son comportement dynamique, qui dépend à la fois des contrôles externes et le la structure interne du modèle de synthèse » [Borin&al.92a][1]. Deux remarques toutefois : d'une part si une relative correspondance peut être établie entre la complexité du signal et la « richesse subjective » du son perçu, il ne peut être établi d'isomorphie rigoureuse entre les deux espaces, d'autre part, il convient de ne retenir dans la complexité dynamique du signal produit par un procédé de synthèse, que ce qui est dû à l'algorithme lui-même et non à la richesse dynamique du flux paramétrique qui le contrôle. Nous laissons quoi qu'il en soit ce critère pour ce qu'il est : vague, difficile à cerner, mais cependant incontournable.

C2 : Quelle est la diversité des sons possibles ? (*)

Ce second critère, plus tangible, peut être en première approche défini comme la capacité d'une technique à produire une catégorie perceptive préalablement choisie, par exemple en référence à n'importe quel son naturel ou son d'instrument traditionnel. Mais il peut se généraliser à la capacité d'engendrer un large éventail de catégories perceptives indépendamment de toute référence. Par ailleurs, certaines techniques ne permettent de produire que des catégories limitées de sons ou se trahissent en tant que techniques à travers une « signature » trop facilement reconnaissable. Cette caractéristique sera également évaluée par ce critère C2.

[1] Traduction libre de l'auteur - *« a good sound is characterized by its complex dynamic behavior, which depends both on external control and on the internal structure of the synthesis scheme »*

Critères relatifs aux modèles mentaux[1]

Ce critère et le suivant abordent deux aspects complémentaires d'une même question, celle de la représentation mentale associée aux différentes techniques. Cette question est en effet décisive car de la représentation mentale va dépendre la plus ou moins grande facilité à intérioriser les possibilités d'une technique et par la même à les exploiter. Mais deux pôles sont à considérer, l'un du côté de ce qui produit, l'autre de celui de ce qui est produit, étant entendu qu'un problème essentiel sera celui de la mise en correspondance des deux.

C3 : Existe-t-il un *modèle mental* simple et opératoire des algorithmes en jeu ?

Le premier des deux critères concerne ce qui produit. Indépendamment des résultats, une technique donnée peut se prêter plus ou moins facilement aux exercices mentaux relatifs à ses règles, à sa combinatoire. Pour les phases purement exploratoires, la facilité de concevoir avant l'expérience effective, les différents possibles facilitera la méthodologie, l'heuristique de l'expérimentation.

La constitution d'un modèle mental efficace par l'utilisateur dépend bien sûr fortement de sa formation (par exemple ses connaissances scientifiques : en traitement du signal, en automatique, en physique, etc.). Elle peut être cependant plus ou moins facilitée par le type de technique selon que celle-ci fait appel à des notions intuitives et communes ou au contraire à des notions très spécialisées.

C4 : Peut-on se représenter correctement *l'espace des sons* synthétisables ?

Pour Borin & al :« un instrument peut être considéré comme une abstraction d'une classe de sons, caractérisée par un timbre, un comportement dynamique et par certaines possibilités expressives [Borin&al.90][2]. Borin & al étendent cette approche aux techniques de synthèse : « nous considérons que le procédé de génération [la technique de synthèse] peut être vue comme une abstraction de la classe des sons produits ; en d'autres termes, les classes d'algorithme déterminent des classes de son, et induisent des classifications parmi elles » [Borin&al.90][3].

Ces remarques posent bien le problème relatif au second aspect de la représentation mentale, celui, pourrait-on dire cette fois, de « l'écoute intérieure ». Mais il est important de souligner que l'abstraction de la classe de sons possibles ne suffit pas à caractériser un instrument. La relation entre la classe de sons et ses dimensions d'une part et leurs parcours par un mode opératoire donné d'autre part vont tout autant déterminer ses potentialités expressives. Néanmoins, la facilité de représentation mentale des classes de sons possibles pour une technique donnée sans obligation d'expérimentation, facilitera le processus de création.

Le critère peut alors consister à estimer si la technique *permet* à l'utilisateur de se construire une représentation mentale des sons synthétisables.

La synthèse des deux présents critères est également importante : une technique donnée peut-elle donner lieu à une représentation mentale aisée de ses règles et modes opératoires ? Permet-elle de se représenter aisément mentalement le champ des possibles sonores ? Et enfin, permet-elle d'associer ou d'apprendre facilement l'association des deux formes de représentations mentales ?

---

[1] La notion de modèle mental est présentée dans la partie II, chapitre 4.

[2] Traduction libre de l'auteur - « *The instrument can be seen as the abstraction of a class of sounds characterized by a timbre, a dynamic behavior and by certain expressive possibilities* »

[3] Traduction libre de l'auteur - « *We believe that the generation process can be seen as an abstraction of the class of sound produced; in other words, classes of algorithms individuate classes of sounds, and induce classifications into them* »

Critère relatif à la modularité

- **C5 : La technique est-elle *modulaire* ?**

Ce critère a une incidence sur critère C3 introduit plus haut.

Dans l'interview qu'il a donnée à Curtis Roads, Max Mathews expose quelques-unes des considérations qui ont présidé à l'élaboration du paradigme des programmes MUSIC[1]. S'expliquant sur le choix d'une modularité maximale, il explique que la modularité lui semblait nécessaire pour offrir à la fois *généralité*, *puissance* et *simplicité* [Roads.80][2]. Il est rejoint en cela par de nombreux auteurs[3].

Nous expliciterons le critère de modularité sous trois angles. Il s'agit d'abord de voir si la technique considérée permet de passer des modèles simples aux modèles complexes par l'adjonction successive de modules supplémentaires. Il s'agit ensuite d'estimer si la technique permet une approche par constructions successives, c'est-à-dire s'il est possible de réutiliser simplement un modèle construit et étudié préalablement dans un modèle plus complexe. Il s'agit enfin d'estimer la prédictibilité de l'approche modulaire, c'est-à-dire la corrélation entre la complexité du son produit et la complexité de la construction modulaire.

Critères relatifs aux paramètres

C6 : Les paramètres sont-ils *intuitifs* ? Ont-il des conséquences sonores identifiables ? (*)[4]

Ce critère est lui aussi à rapprocher des critères C3 et C4. Il en est le pendant quantitatif, si l'on considère que les constructions structurelles des algorithmes sont des déterminations qualitatives.

La psychoacoustique a mis en évidence plusieurs dimensions de la perception auditive ou « paramètres perceptifs » : sonie, hauteur tonale et spectrale, brillance par exemple. Certaines techniques tendent à offrir un contrôle indépendant des dimensions de la perception[5]. Avec d'autres, la modification d'un paramètre aura des conséquences sur plusieurs paramètres perceptifs ; il s'agit alors d'estimer si la manière dont se projettent les paramètres sur les dimensions de la perception est pertinente et intuitive.

---

[1] Le paradigme des programmes Music (Music V, C-Sound…) est une référence majeure de la synthèse des sons. Notons en outre que Music III (1959), de par les concepts de générateur élémentaires (*unit generator*) et de blocs de construction (*building blocks*) qu'il a introduit, est parfois considéré comme le premier programme modulaire, préfigurant la programmation objet [Risset.99b].

[2] Citation originale : *"I wanted to give the musician a great deal of power and generality in making the musical sounds, but at the same time I wanted as simple a program as possible ; I wanted the complexity of the program to vary with the complexity of the musician's desires"* [Roads.80].

[3] Entre autre par Borin & al. Citation originale : *"We have to consider that, when a model is used in Computer Music compositions, its aim is to produce musicaly interesting sounds. This requires flexibility in the model design and versatility in the choice of parameters.* We believe that the modularity in the structure of the model is a necessary characteristic of a musican-oriented model*"* [Borin&al.90].

[4] En reprenant les mots de Jaffe, qui propose un critère similaire : *"this topic concerns whether the parameters map in an intuitive manner to musical attributes like musical dynamics and articulation, or whether they are mathematical variables with very little correlation to real-world perceptual or musical experience"* [Jaffe.95].

[5] Un axe de recherche intéressant consiste à rechercher, pour un algorithme de traitement ou de synthèse, une base paramétrique "intuitive", puis à effectuer un changement de base entre paramètres de l'algorithme et paramètres proposés à l'utilisateur (voir à ce sujet, par exemple, les recherches menées à l'IRCAM sur le contrôle « spatialisateur » de son, [Jot.95]). La corrélation entre paramètre des algorithmes et paramètres perceptifs peut toutefois être une propriété plus ou moins native de la technique.

C7 : Les paramètres se *comportent*-ils correctement ? (*) Sont-ils *prédictibles* ?

Jaffe introduit un critère spécifique relatif à la linéarité de la relation entre les paramètres et leurs effets sensibles. Il indique notamment qu'un paramètre dont les variations influent très peu sur le son, ou inversement modifient le timbre de façon très importante paraîtra peu satisfaisant : « idéalament, une modification d'un paramètre entraînera un transformation proprtionnelle dans le son [Jaffe.95][1].

De notre point de vue, le plus important est bien la prévisibilité de l'effet des changements paramétriques, étant entendu qu'un paramètre peu prévisible (dont l'effet est non-linéaire par exemple) nuit, en général, à l'efficacité de la technique considérée. Ce critère, incidemment, prendra souvent le pas sur le précédent. Il convient néanmoins, encore une fois, de ne pas prendre trop en absolu ce type de considérations. En effet, l'existence de seuils catégoriels ou de non-linéarités fortes entre l'évolution d'un paramètre et celle de ses effets peut jouer un rôle heuristique fondamental. D'une manière générale, les relations entre les causes et les effets (qu'il s'agisse de leurs caractéristiques qualitatives ou quantitatives) n'est pas simple et apprendre ces relations est parfois plus productif que de tenter de les homogénéiser ou banaliser à travers des artifices techniques qui peuvent s'avérer très lourds.

C8 : L'identité des modèles (et des sons) est-elle *robuste* face aux variations paramétriques ? (*)

Il s'agit d'estimer si le comportement des modèles est robuste face aux variations paramétriques, c'est-à-dire si les sons synthétisés conservent une identité marquée dans le cas de *variations* d'un ou plusieurs paramètres.

On peut considérer ce critère comme un corollaire des précédents. Jaffe lui attribut cependant une importance particulière. Il explique que « Pour obtenir un instrument musical - qui soit plus qu'une photo d'une situation particulière - il est essentiel de pouvoir synthétiser l'expressivité, c'est-à-dire une variété importante dans le contexte d'un son particulier ». Si de nombreuses techniques de synthèse permettent d'obtenir très précisément des notes isolées, ce critère montre qu'elles doivent aussi permettre une famille de sons proches, et qui évoquent une variation expressive autour d'un caractère sonore *robuste.*

Critères relatifs aux flux dynamiques et à la situation de « jeu »

C9 : Les flux dynamiques prévus par la technique permettent-ils un « contrôle » précis expressif et intuitif ? (*)

Ici, il s'agit ici d'estimer la pertinence des flux (de paramètres) dynamiques - étant entendu que les flux peuvent être temps-réel ou temps-différé – c'est à dire l'intérêt de la technique en situation de jeu. Ce point fait écho aux critères C6 à C8 consacrés aux paramètres, puisqu'il s'agit à nouveau de s'interroger sur la précision et la prévisibilité de l'effet des flux. Il nécessite une formulation indépendante tant le contrôle dynamique des processus de synthèse pose des questions autres que la conception des modèles et le réglage de leurs paramètres.

[1] Traduction de l'auteur - *"Ideally, a change in a parameter produces a proportional change in the sound"*.

Certaines techniques laissent la possibilité d'un contrôle en flux de tous les paramètres. La largeur du flux paramétrique est alors très importante. Il peut être difficile de choisir quels paramètres faire varier, et plus ardu encore de déterminer les valeurs dans le temps de chacun d'entre eux pour obtenir un effet escompté – surtout si la technique est peu satisfaisante au regard des critères C6 à C8. Pour y parvenir, on peut envisager de mettre en œuvre une couche logicielle supplémentaire qui établirait des corrélations entre paramètres du modèle et calculerait à chaque instant l'ensemble de leurs valeurs à partir d'un nombre restreint de « paramètres de contrôle » de plus haut niveau. On touche alors aux problématique du contrôle de synthèse, de la *synthèse par règle* (voir [Barrière.90], [Rodet&al.90] par exemple) ou encore du *lien instrumental* [Wanderley.99]. Si une telle approche a montré son intérêt, il reste qu'elle suppose l'ajout d'un étage à celui de la technique de synthèse elle-même, étage qui a sa propre logique. La prise en main de la technique de synthèse (par un musicien par exemple) est rendue plus difficile puisqu'en plus des algorithmes de génération du son il faut encore maîtriser ceux du lien instrumental.

D'autres techniques vont imposer certains flux de paramètres, c'est-à-dire orienter ou restreindre par nature les types de données susceptibles d'être échangées pendant la phase de jeu. Ces techniques ont en commun de coder les possibilités de jeu - i.e. les variations dynamiques possibles, pendant le jeu, sur un seul instrument - dans la structure même de l'algorithme. Elles peuvent être préférables dans la mesure où elles incorporent en propre la problématique du contrôle dynamique.

C10 : Les types de flux en entrée et en sortie sont-ils divers ? La technique est-elle "ouverte" ?

Le son synthétisé est un premier flux indispensable, mais d'autres flux sont envisageables - et nécessaires. Il s'agit alors d'estimer si la technique est ouverte, c'est-à-dire si elle est capable de "collaborer" avec d'autres ou avec des périphériques. Il faut donc déterminer les types de flux prévus par la technique en entrée et en sortie, ainsi que la largeur de bande du contrôle paramétrique.

En entrée, on considérera le débit et les types de données prévus (nativement) par la technique. Il importe de rappeler que le débit en entrée doit être inférieur au débit en sortie, sans quoi il n'y a plus synthèse mais transformation. En revanche, il est nécessaire que le débit soit important pour qu'une certaine expressivité soit possible.

Pour la sortie, on regardera si la technique est susceptible de générer des signaux utilisables pour une finalité autre que sonore, par exemple pour commander d'autres synthèses ou des programmes visuels. On s'intéressera enfin à la possibilité de mettre en place une communication bidirectionnelle, indispensable dès lors qu'on veut installer une situation *d'interaction* (gestuelle par exemple).

Critère relatif à l'efficacité algorithmique

C11 : L'algorithme est-il efficace ? (*)

En situation temps-réel, le coût algorithmique détermine la complexité des modèles et le nombre de voies calculables en parallèle. En situation temps-différé, il influe sur le nombre d'itérations que pourra effectuer le musicien durant la mise au point de son modèle et de ses paramètres. L'efficacité algorithmique varie beaucoup d'une technique à l'autre,

pour une richesse sonore équivalente, et son estimation constitue un critère important. Ce critère a joué un rôle décisif dans le succès de certaines techniques.

L'efficacité d'une technique ne s'évalue pas seulement suivant l'importance des calculs nécessaires à la synthèse ; la quantité de mémoire utilisée et la quantité d'information véhiculée par les flux dynamiques doivent aussi être considérés.

A la notion de coût algorithmique se rattache, enfin, celle de *latence irréductible*[1]. Certains algorithmes, pour élaborer les échantillons de sortie, nécessitent par construction l'acquisition préalable de données en entrée. Ils introduisent en conséquence un temps d'élaboration irréductible des échantillons en sortie. Ce délai posera problème pour une implémentation proche du temps réel. La latence est nulle si le premier échantillon en sortie apparaît immédiatement après la réception des données en entrée (signal échantillonné ou évènement). Au contraire, les algorithmes par transformation (voir paragraphe suivant) présentent souvent une latence importante.

Critères relatifs à l'environnement

Nous envisageons finalement deux critères qui dépassent le seul cadre des algorithmes de génération du son pour s'intéresser aux programmes qui les complètent, c'est à dire à l'environnement logiciel de la technique considérée.

C12 : Existe-t-il des algorithmes de génération de modèles par analyse d'un existant, et notamment par analyse d'un son ? (*)

Le premier est relatif aux outils d'analyse permettant l'élaboration automatique d'un modèle pour la re-synthèse d'un existant : signal sonore, ensemble de sons, mesures physiques sur un instrument réel ou tout autre ensemble de mesures d'un phénomène que l'utilisateur souhaiterait pouvoir re-synthétiser – avec une modification éventuelle.

Certaines techniques se prêtent difficilement à une telle projection du « réel » sur le « synthétique ». D'autres, au contraire, l'autorisent par nature - certaines d'entre elles étant même irréductiblement liées à une démarche d'analyse / synthèse.

C13 : Existe-t-il un environnement convivial pour la mise en œuvre de la technique ?

Pour qu'un musicien puisse prendre en main une technique de synthèse, il faut qu'un environnement logiciel soit disponible. L'organisation d'un tel environnement dépend de manière importante de la technique qu'il interface, donc de son comportement face aux critères précédents (et notamment C3, C6, C9, C11). L'environnement ajoute cependant des fonctionnalités supplémentaires, non natives à la technique, de sorte qu'une technique de synthèse, lorsqu'on l'envisage comme un moyen pour la création, ne peut finalement être considérée indépendamment de son environnement logiciel. Il convient alors d'estimer si l'environnement permet de contourner certains de ses défauts (relatifs à C3, C5, C6, C7 par exemple), et de manière générale permet une créativité accrue.

On voit là que ce dernier critère est très général et que, comme le critère C1, il dépend drastiquement du contexte d'utilisation et des objectifs de l'utilisateur. L'évaluation des environnements de synthèse nécessiterait une autre série de critères – à supposer qu'elle soit possible. Il nous a semblé important de l'évoquer malgré sa généralité. Incidemment, sa satisfaction relativement à CORDIS-ANIMA, est au cœur de ce document.

[1] Jaffe présente la latence de l'algorithme sous la forme d'un critère indépendant. Il nous a semblé plus intéressant de l'introduire comme un aspect de l'efficacité algorithmique.

2.2. Evaluation globale des modèles physiques en regard des critères

Ce paragraphe propose une évaluation *globale* des modèles physiques relativement aux 13 critères que nous avons précédemment retenus[1].

Le premier critère maximisé lorsqu'on met en œuvre une technique par modèles physiques est celui de la richesse musicale des sons synthétisés (C1). Ces techniques offrent une identité sonore des modèles particulièrement robuste face aux variations paramétriques (C8) ; elles sont donc intéressantes par leur aptitude à la création de familles de modèles, et parallèlement de familles de sons pertinentes. Les paramètres des modèles physiques sont en général intuitifs (C6) : s'ils ne sont pas directement ancrés dans les dimensions aujourd'hui connues de la perception, les effets sonores résultants de leurs variations peuvent être anticipés intuitivement grâce à une culture commune résultant de nos actions journalières. C'est cependant dans un contexte de jeu que les paramètres des modèles physiques rencontrent tout leur intérêt (C10). Les flux à contrôler (force, position, vitesse...) constituent en eux-mêmes une représentation des actions entreprises par un musicien dans une situation de jeu traditionnelle, et l'effet d'une variation sur un tel paramètre dynamique est plus immédiatement naturel et prédictible qu'avec des paramètres de modèles de signaux. Les techniques par modèles physiques offrent des éléments de réponse pertinents à la question du contrôle de synthèse.

A l'inverse cependant, les techniques par modèles physiques sont coûteuses (C11), ce qui n'est guère étonnant puisqu'elles simulent « plus » que le phénomène sonore. Les types de flux sont, à priori, restreints par les techniques par modèles physiques (*via* les notions de force et de position), de sorte qu'elles pourraient être considérées à priori comme moins « ouvertes » que les techniques relevant d'une approche signal – rappelons qu'on peut voir là tout aussi bien un gain en ce qu'elles différencient les paramètres responsables de la permanence du modèle des paramètres de jeu. Enfin, en règle générale, l'approche par modèles physiques de la synthèse de son se prête mal à la mise en œuvre d'algorithmes permettant la genèse automatique d'un modèle à partir d'un existant, notamment d'un son[2].

Les autres critères (modèle mental, comportement des paramètres, diversité des sons, représentation de l'espace des sons, modularité, environnement) sont quant à eux variables suivant les techniques.

---

[1] Notons qu'on pourrait également confronter chacune des techniques de synthèse en « modèles de signaux » à cette base critère (voir [DePoli.83] pour une description des algorithmes concernés). Quelques exemples. La *synthèse par tables d'ondes* est particulièrement intéressante de par son *efficacité* (C11). La *synthèse additive* permet une bonne représentation mentale des algorithmes en jeu (C3), est particulièrement modulaire (C5), offre des paramètres intuitifs (C6) et permet un un bouclage Analyse / Synthèse efficace (C12). Elle tend cependant à imposer une signature marquée au sons (C2) et suppose que toutes les variations dynamiques soient codées dans le flux paramétrique (C9). La synthèse par *modulation de fréquence* (synthèse FM) est très efficace (C11) tout en offrant des sons riches (C1). Par contre, elle est particulièrement difficile à mettre en œuvre (C3, C5, C6, C7), tend à imposer sa signature (C2) et ne permet guère d'algorithme de re-synthèse (C12). Les autres techniques de synthèse (synthèse granulaire, modulation en anneau, synthèse source-filtre ou *soustractive*, etc.) pourraient, de même, être confrontées au critères proposés. Nous préférons nous consacrer plus en détail aux techniques de modélisation physique.

[2] Cela doit être nuance, dans la mesure où de nombreuses recherches vont en ce sens - voir par exemple la notion de *génération* dans le corps du document, chapitre 18, et [Szilas&Cadoz.93].

2.3. Des critères spécifiques aux modèles physiques

Les attentes relatives aux modèles physiques sont suffisamment distinctes de celles entretenues vis-à-vis de la synthèse pour qu'une évaluation des techniques « musicales » de modélisation physique ne puisse être conduite de façon exhaustive sur la base des 13 critères précédemment introduits. Aussi, dans ce paragraphe, nous proposons quelques nouveaux critères qui, additionnés aux précédents, permettent de mieux distinguer les potentiels respectifs de chaque approche du « modèle physique ».

Critères relatifs au phénomène sonore

Nous envisageons d'abord trois critères relatifs aux phénomènes sonores générés par les modèles physiques lors de leur simulation.

C'1 : Précision des phénomènes générés

Selon Borin & al, l'aptitude des techniques de modélisation physique pour la synthèse des sons d'instruments réels reste l'un des critères fondamentaux pour leur validation :

> « Les modèles physiques s'appuient sur le monde réel non seulement lorsqu'il s'agit de trouver une inspiration pour concevoir un modèle conceptuel, mais aussi lorsqu'il s'agit d'identifier les paramètres des modèles et, plus important, d'évaluer les résultats. […] Les résultats sonores de n'importe quelle technique sont souvent comparés aux sons naturels, mais les modèles physiques ont la caractéristique de considérer cette référence comme une hypothèse de validation » [Borin&al.92a][1].

Notre premier critère sera donc relatif à la *précision* des phénomènes générés, qui s'évalue en confrontant simplement les sons de l'instrument réel à modéliser avec les sons de synthèse. Ce critère est décisif lorsque l'objectif est *mimétique* et qu'on souhaite remplacer le référent par son simulacre – dans un synthétiseur par exemple[2]. Il est moins important dès lors qu'on espère une bonne « plausibilité » des sons, mais qu'on ne cherche pas précisément à reproduire les sons d'un instrument particulier.

C'2 : Diversité des métaphores possibles

Ce second critère renvoie au critère C2 relatif aux classes de sons synthétisables ; nous le précisons ici en faisant explicitement référence à l'instrumentarium réel, du fait à nouveau des vertus « mimétiques » des modèles physiques. Il s'agit ici d'estimer les catégories d'instruments qu'il est possible d'évoquer avec la technique de modélisation considérée. Certaines techniques sont dédiées à la modélisation de certaines catégories d'instruments (les vents ou les cordes par exemple). D'autres, plus générales (mais parfois moins « précises »), permettent de représenter un plus grand nombre des instruments de musique. Ces dernières techniques seront parfois préférables, par exemple lorsqu'on veut, au sein d'un seul environnement, permettre l'évocation d'une importante diversité de timbres.

---

[1] Traduction libre de l'auteur - *"Physical models refers to real world not only to find inspiration for building conceptual models, but also to identify system implementation parameters and, most importantly, to evaluate the results. (...) Results are often compared with natural sounds for qualitative judgments during synthesis using any technique. But Synthesis by PM has the unique feature of taking this reference as its validating hypothesis"*

[2] Ce critère reste difficile à majorer. Aujourd'hui encore, par exemple, les meilleurs *pianos numériques* ne sont pas basés sur des modèles physiques, mais plus sur les techniques plus traditionnelles du *sampling*. De façon plus générale, des modèles de signal peuvent souvent permettre l'obtention d'une excellente *précision* des sons.

Critère phénoménologique étendu

C'3 : Diversité et cohérence des phénomènes engendrés

Nous avons indiqué que les modèles physiques ne sont pas à priori limités à la génération de phénomènes sonores. Cette propriété n'est en fait pas commune à toutes les approches, certaines d'entre elles étant par nature orientées vers la modélisation des comportements impliqués dans la genèse des *sons*. Il s'agit alors de préciser les catégories de phénomènes (autres que sonores) que peuvent engendrer les modèles, ainsi que la cohérence de ces divers phénomènes. Il s'agit, de façon corollaire, de déterminer si la technique est exclusivement *sonore* ou si elle permet la conception « d'objets » plus divers et leur perception par d'autres modalités sensorielles.

Ce critère nous semble essentiel, quand bien même l'objectif reste musical. Plusieurs travaux – dont ceux menés au sein du laboratoire – montrent en effet l'importance de la *collaboration* des sens humains dans les situations instrumentales. Lorsque la technique l'autorise, les phénomènes sonores, visuels et/ou gestuels générés de façon cohérente participent à une perception étendue du modèle simulé et soutiennent la qualité du jeu de l'instrumentiste. Par ailleurs, nous avons évoqué dans le document la possibilité d'utiliser des modèles physiques pour engendrer des *formes* musicales – avec des modèles basse fréquence par exemple – et non plus seulement des sons ; la possibilité d'une telle démarche est prise en compte par ce critère.

Critères pour la mise en œuvre de la technique

Nous proposons donc deux critères pour évaluer la facilité de mise en œuvre des techniques.

C'4 : Robustesse de la plausibilité

Lors d'une processus de modélisation, le musicien ne mettra en œuvre les connaissances scientifiques du physicien ou du mathématicien, mais adoptera une démarche intuitive, exploratoire, empirique. Certaines techniques, lorsqu'elles sont pratiquées avec une telle démarche, permettent difficilement l'obtention de modèles qui génèrent des phénomènes – notamment sonores – « plausibles ». Il convient alors d'estimer la *robustesse* de la technique. En d'autres termes, il s'agit d'estimer si la plausibilité des sons est une propriété qu'on obtiendra aisément avec la technique considérée et si elle perdure quand un modèle est modifié.

C'5 : Efficacité et simplicité du modèle mental – « impression de réalité »

Ce critère étend le critère C3 proposé précédemment, dans la mesure ou avec les modèles physiques, la possibilité d'un modèle mental des algorithmes en jeu prend à la fois une importance accrue et un sens différent. Nous considérons que le modèle mental attaché à une technique par modèle physique sera d'autant plus intéressant qu'il permet de considérer et manipuler les modèles *comme des objets réels* – quand bien même ils sont représentés – et non pas comme des ensembles d'équations ou des constructions théoriques. Il convient, en d'autres termes, d'estimer *l'impression de réalité* qu'offre le modèle mental associé à la technique considérée.

Notion de profondeur de la modélisation

Nous conclurons cette partie en évoquant la notion de *profondeur* de la modélisation. Claude Cadoz distingue trois types possibles de modèles : phénoménologiques, fonctionnels et structurels [Cadoz.90a][1]. Une courbe expérimentale ou un enregistrement sont, par exemple, des modèles *phénoménologiques*. Lorsqu'on trouve une fonction d'interpolation (resp. un modèle de signal) permettant la reconstruction de la courbe ou la synthèse du son, on dispose d'un modèle *fonctionnel*. Lorsque, par contre, on considère non pas le phénomène mais sa cause (le corps sonore par exemple), qu'on la décompose en éléments plus simples en interaction et que pour chacun d'eux on propose un modèle, on met en œuvre une démarche de modélisation *structurelle*[2].

C'6 : profondeur de la modélisation

Partant de cette catégorisation, on peut définir une notion de profondeur des modèles, en disant que plus l'analyse structurelle est fine et poussée (i.e. : plus les « éléments » sont simples), plus le modèle est « profond »[3]. C'est sur la base de cette définition que nous introduisons notre sixième et dernier critère, qui consiste précisément à évaluer la profondeur de la technique considérée.

La profondeur des techniques de modélisation est un critère particulièrement important, qui englobe d'ailleurs pour partie les précédents. S'il n'est pas à priori nécessaire de disposer d'un modèle profond pour obtenir une grande précision phénoménologique[4], en particulier pour des événements isolés ou singuliers. Cela devient en revanche déterminant dès lors que l'on envisage les éléments dans un contexte, contexte multisensoriel par exemple, ou contexte dynamique dans lequel des ensembles d'événements sont mis en relation ou en confrontation. Ce n'est en effet souvent que dans ces situations que la cohérence (ou l'incohérence) « profonde » des modèles se révèle et traduit en explicite une richesse (ou une absence de richesse) structurelle intrinsèque.

Par ailleurs, un modèle profond favorise une représentation mentale plus aisée des algorithmes en jeu, dans la mesure ou les briques élémentaires avec lesquelles le modèle est construit sont alors plus simples et plus faciles à appréhender.

---

[1] Ce texte est reproduit *in-extenso* à l'annexe C.

[2] Il convient de remarquer que les « modèles élémentaires » constitutifs d'un modèle structurel sont eux phénoménologiques ou fonctionnels. La profondeur, c'est précisément l'endroit où l'analyse structurelle s'arrête. Notons que lors de la simulation d'un modèle structurel les phénomènes *émergent* des interactions qui ont lieu « en profondeur » entre éléments.

[3] La notion de profondeur ne doit donc pas être confondue avec la notion de précision du modèle, ou encore de complexité des phénomènes considérés. Un modèle de propagation qui prendrait en compte le couplage entre ondes transverses, longitudinales et de torsion par exemple ne serait pas nécessairement un modèle profond au sens ou nous l'entendons ici, mais plus un modèle « précis » (critère C' 1).

Pour Cadoz, incidemment, un modèle « physique » *est* précisément un modèle « profond » : un modèle peut être dit « physique » si il existe une correspondance entre constituants du référent et constituant du modèle. Avec cette approche, plus cette correspondance est poussée dans les constituants élémentaires du référent, plus le modèle est « physique ».

[4] Pour preuve, à nouveau, les excellents modèles de piano pourtant basés sur le sampling.

Critères généraux pour les techniques de synthèse

Critères relatifs aux sons productibles

C1	La technique permet-elle la synthèse de sons « musicalement riches » ?
C2	Quelle est la diversité des sons possibles ? (*)

Critères relatifs aux modèles mentaux

C3	Existe-t-il un *modèle mental* simple et opératoire des algorithmes en jeu ?
C4	Peut-on se représenter correctement *l'espace des sons* synthétisables ?

Critère relatif à la modularité

C5	La technique offre-t-elle une *modularité* importante et intuitive ?

Critères relatifs aux paramètres

C6	Les paramètres sont-ils *intuitifs* ? Ont-il des conséquences sonores identifiables ?
C7	Les paramètres se *comportent*-ils correctement ? Sont-ils *prédictibles* ?
C8	L'identité des modèles (et des sons) est-elle *robuste* face aux variations paramétriques ?

Critères relatifs aux flux dynamiques et à la situation de « jeu »

C9	Les flux dynamiques prévus par la technique permettent-ils un « contrôle » précis expressif et intuitif ?
C10	Les types de flux en entrée et en sortie sont-ils divers ? La technique est-elle "ouverte" ?

Critère relatif à l'efficacité algorithmique

C11	L'algorithme est-il efficace ?

Critères relatifs à l'environnement

C12	Existe-t-il des algorithmes de génération de modèles par analyse d'un existant, et notamment par analyse d'un son ?
C13	Existe-t-il un environnement convivial pour la mise en œuvre de la technique ?

Critères propres aux modèles physiques

Critères relatifs au phénomène sonore

C'1	Précision des phénomènes générés : quelle est la fidélité des phénomènes générés (par rapport à ceux que génère l'objet réel) ?
C'2	Diversité des métaphores possibles : quelle(s) classes(s) d'objet ou d'instrument peuvent être modélisés ?

Critère phénoménologique étendu

C'3	Diversité et cohérence des phénomènes engendrés : est-ce que la technique est orientée vers la genèse du son, ou permet-elle d'engendrer d'autres phénomènes ?

Critères pour la mise en œuvre de la technique

C'4	Robustesse de la plausibilité : les phénomènes restent-ils physiquement plausibles dans le contexte de variations ?
C'5	Efficacité et simplicité du modèle mental – « impression de réalité » : le raccourcis permettant d'envisager les modèles comme des « objets » virtuel est-il possible ?

Notion de profondeur de la modélisation

C'3	Profondeur de la modélisation : la technique permet-elle une démarche de modélisation structurelle poussée ?

Figure 2: tableau résumé des critère d'évaluation comparée

3. Principales techniques de modélisation physique pour la musique

Dans ce dernier chapitre, nous introduisons brièvement les principales techniques de modélisation aujourd'hui pratiquées en Informatique Musicale, et nous les confrontons aux critères que nous venons d'exposer.

3.1. Analyse numérique, éléments finis, différences finies... Les outils de l'acousticien

Pour beaucoup, la modélisation physique dans l'environnement informatique est indissociable des techniques de discrétisation numérique. Dès lors, pour parvenir à l'étape de simulation, il faut d'abord formuler un modèle suivant les lois de la Physique – Newtonienne, des fluides, etc. – sous la forme d'un ensemble d'équations qui traduisent le fonctionnement de chaque partie de l'instrument (résonateur, corde, etc.) ainsi que leurs interactions. La résolution algébrique de ce système à temps et espace continus est cependant souvent impossible – du fait notamment des non-linéarités. Pour *simuler* le modèle, il faut donc avoir recours à des techniques numériques de discrétisation. telles que les différences finies ou les éléments finis[1]. Un second modèle est obtenu, *approximation plus ou moins fidèle du premier* (on parle alors de divergence ou biais numérique) mais qui a la particularité d'être « calculable » dans l'univers discret de l'ordinateur.

Nous qualifierons de *traditionnelle* une telle approche en ce qu'elle fut la première utilisée et qu'elle repose sur une analyse de l'objet à modéliser suivant les lois traditionnelles de la physique.

L'approche traditionnelle reste principalement l'apanage des acousticiens désireux d'étudier et comprendre la physique difficile des instruments de musiques. Elle constitue essentiellement une technique de *validation*, précise certes (la précision dépendant en fait de celle du modèle « continu » et de la technique de discrétisation employée), mais qui suppose pour être pratiquée un bagage scientifique et technique conséquent. Elle est particulièrement peu *modulaire* (C5), et permet difficilement la mise en œuvre d'un processus de création basée sur l'acte de modélisation. Le *modèle mental* qui lui est attaché est trop peu intuitif pour que des musiciens puissent la maîtriser.

Pour autant, des approches par *discrétisation* restent utilisées en musique. Elles peuvent être prises en main par des musiciens dès lors que les équations mathématiques *et* la technique numérique employée leur sont cachées. C'est le cas des deux techniques que nous détaillons ci-après : l'approche dite « classique » et celle des guides d'ondes.

3.2. L' "approche classique" des travaux de Ruiz et Hiller

En 1971, Ruiz et Hiller sont les premiers à proposer un modèle physique dans le but explicite de synthétiser des sons à vocation musicale, et non plus seulement d'étudier la physique des corps sonores. Leur approche reste cependant profondément inscrite dans la démarche traditionnelle : la spécification de la forme et des constantes physiques de l'objet, des conditions aux limites, de l'état initial et des caractéristiques d'un système excitateur permet l'écriture d'un système plus ou moins complexe d'équations aux dérivées partielles, qu'il s'agit de résoudre par une technique d'approximation numérique.

[1] La technique numérique des « éléments finis » est probablement la plus utilisée, à tel point que pour beaucoup la simulation physique s'identifie au calcul par éléments finis - en tout cas lorsque l'objet à modéliser est solide.

La spécificité de l'approche de Ruiz et Hiller tient au choix qu'ils effectuent pour la résolution approchée des équations ; ils proposent pour ce faire un paradigme à masses et ressorts dans un espace tridimensionnel, puis étudient la divergence induite par une telle discrétisation par rapport au modèle continu, supposé idéal[1].

Selon Jean-Marie Adrien, « les limites de l'approche de Ruiz et Hiller résultent du fait que les solutions analytiques sont de manière générale arbitraires et insuffisantes à l'oreille ou trop complexes et inutilisables en pratique. Les auteurs négligent donc les possibilités de leur modèle en matière de contrôle et réduisent ses performances en matière de transitoire » [Adrien.89].

- Le principe du synthétiser TAO (*TAO music synth'*)

Le système TAO, développé depuis 1995 en Grande Bretagne par Mark Pearson ([Pearson.95], [Pearson.96], [TAO]) est une mise en œuvre récente de la démarche initiée par Hiller et Ruiz. Il est constitué d'un moteur de simulation et d'un langage. Les modèles sont définis par la donnée de la forme et de la taille des différents objets, de quelques paramètres de nature physique (raideur, viscosité), des conditions aux limites et de la description des connexions entre les objets. Si la technique de discrétisation/simulation des objets relève bien d'une discrétisation en masses et ressorts, ces "cellules" ne sont pas directement disponibles à l'utilisateur. Les interactions entre objets relèvent qui plus est d'un formalisme différent.

3.3. Les guides d'ondes

Approche théorique

En 1979, l'algorithme Karplus-Strong a fait date en ce qu'il permettait la synthèse temps réel de plusieurs sons de cordes pincées en parallèle :

> "L'algorithme de Karplus-Strong a montré, à la surprise générale, que le modèle physique d'une vraie corde vibrante pouvait être ramené à une moyenne sur deux points sans multiplication tout en donnant des résultats musicalement utiles." [Smith.96].

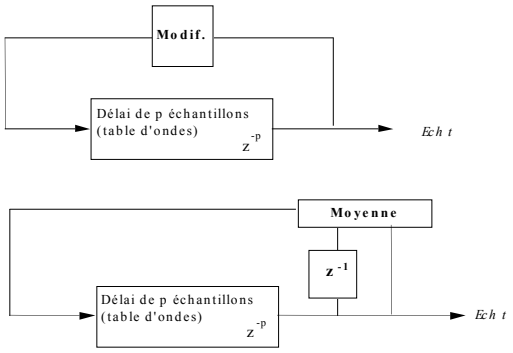


Figure 3 : l'algorithme Karplus-Strong

L'algorithme consiste à faire évoluer en cours de calcul une table d'onde initialisée aléatoirement, en modifiant l'échantillon en sortie de la ligne à retard avant sa réinsertion en tête [Karplus&Strong.83]. La modification la plus simple est réalisée en effectuant la moyenne de l'échantillon t et de l'échantillon t-1 ; c'est elle qui permet la synthèse de sons de cordes pincées réalistes.

Le signal résultant est une suite récurrente d'ordre p :

$$Y_t=(1/2)(Y_{t-p}+Y_{t-p-1})$$

($y_0 ... Y_p$ aléatoires)

[1] On est tenté d'établir un parallèle avec le formalisme CORDIS-ANIMA. Outre que les deux approches diffèrent par leur genèse (et sans rentrer dans le détail, par leur algorithmique), elles portent deux vision différentes de la modélisation. D'un côté le modèle simulé approxime un modèle continu supposé idéal, de l'autre le choix est de placer la discrétisation au cœur du système de représentation. Enfin, elles placent le musicien-compositeur et le principe de la modélisation dans des interactions différentes, et supposent deux démarches de modélisation radicalement opposées : dans le premier cas, le modèle est continu puis discrétisé, dans le second il est construit par la connexion de modules élémentaires. Ces deux approches induisent ainsi deux processus de création (si l'on considère que cette conception des modèles est du ressort du créateur) radicalement différents.

Cette récurrence induit un temps d'amortissement de la $n^{ème}$ harmonique en p^3/n^2, cause du caractère sonore de l'algorithme qui évoque correctement le son des cordes pincées.

Il convient de remarquer que la genèse de l'algorithme ne s'appuie sur aucune considération physique, pas plus que son implémentation qui relève des techniques traditionnelles du filtrage numérique. Ce n'est que plus tard que Julius O. Smith en a proposé une interprétation physique. Il l'a alors généralisé en définissant le principe des *guides d'ondes* que nous connaissons aujourd'hui.

Un guide d'onde monodimensionnel est constitué d'une double ligne à retard bouclée aux extrémités et dont les pertes sont « condensées » aux points d'accès. Il peut être implémenté à l'aide de filtres numériques simples.

Un tel banc de filtres numériques réalise une discrétisation de *l'équation de propagation aux dérivées partielles* bien connue des acousticiens. De fait, la mise au point d'un guide d'onde pour la simulation du résonateur d'un instrument est en général consécutive à l'étude des équations de propagation correspondante suivant les arguments traditionnels de la physique. Ainsi, les guides d'ondes permettent la simulation du *phénomène* de propagation des ondes dans la matière, et donc de modéliser les parties résonnantes des instruments considérées linéaires.

Les recherches plus récentes, en matières de guides d'ondes, portent sur les modalités de leur excitation – le guide d'onde ne représentant que la partie résonante – sur les interactions possibles entre différents guides – permettant par exemple la simulation du couplage entre ondes longitudinales et transversales dans la matière – et sur les réseaux bi ou tri-dimensionnels de filtres ([Smith.96], et de façon générales les actes des récentes conférences International Computer Music Conference, ICMC).

La technique des guides d'ondes a pris une importance prépondérante ces dernières années, tant scientifique que commerciale (synthétiseur VL1 de Yamaha par exemple). Les raisons techniques de ce succès tiennent en premier à leur efficacité algorithmique : les filtres à retard qui les implémentent sont bien connus et très rapides. Les guides d'ondes sont par ailleurs *précis* (C'1), du moins dans le cadre de leur domaine d'application (i.e. : pour la modélisation des résonateurs supposés linéaires). On peut considérer les guides d'ondes comme des modèles relativement profonds, mais il faut noter que le « module élémentaire » qu'est le filtre à retard n'est pas un modèle de la *matière* mais un modèle des ondes qui s'y propagent. La *diversité phénoménologique* (C' 3) des guides d'ondes est en conséquence limitée : basés sur la représentation du phénomène de propagation, ils sont nettement orientés vers les modèles sonores. De même, et pour la même raison, ils sont peu adaptés à la simulation d'objets hautement non-linéaires (*diversité sonore* C'2). Enfin, il faut noter que la conception d'un modèle en guides d'ondes ne peut que difficilement être prise en charge par un musicien non–physicien (*modèle mental* C'5). En général, un modèle préconçu lui est proposé, et seules lui sont en général accessibles des variations paramétriques sur des macro-modules préalablement construits et des connexions entre ces modules.

3.4. L'approche source / filtre en interaction, techniques numériques non linéaires

Les instruments de musique sont pour la plupart constitués d'un corps résonnant excité de façon plus ou moins complexe. La « source » (i.e. ici l'origine physique) de l'excitation est généralement assez simple : on peut par exemple considérer que la vitesse d'un archet ou que la pression buccale d'un clarinettiste sont constantes. L'excitation résulte alors d'une *interaction*, essentiellement non-linéaire, entre « *source* » et *résonateur*.

Cette *analyse structurelle* a conduit à proposer des algorithmes constitués d'un filtre numérique[1] et d'une source de signal, les deux étant couplés à l'aide d'une fonction d'interaction (ou de rétroaction) non linéaire. Le modèle, de par sa structure, est dès lors susceptible de transformer une source de signal élémentaire (impulsions, signal constant…) en oscillations sonores. On parle alors d'un modèle source/filtre en interaction [Rodet&al.90], [Rodet&Vergez.96].

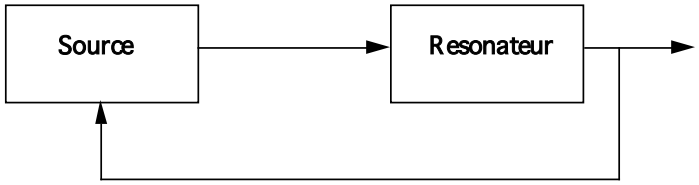


Figure 4 : l'approche source filtre en interaction

On comprendra qu'un modèle source/filtre en interaction reste à la limite de la « modélisation physique », dans la mesure où elle repose principalement sur les arguments issus du traitement du signal. L'approche est par ailleurs peu profonde (C'6) : si elle découle d'une étude de la structure des instruments, celle-ci reste élémentaire. Elle se place dans la continuité de la synthèse soustractive. De ce fait, d'ailleurs, la représentation mentale de ses algorithmes peut être efficace pour tous ceux qui connaissent les principes du traitement du signal. Elle ne permet guère cependant à l'utilisateur de se construire une « impression de réalité » vis à vis des modèles qui en découlent (C'5). Enfin, l'approche n'offre guère de diversité (C'2), ou pour être plus précis un modèle doit être étudié avec attention pour chaque type d'instrument, et n'entretiendra que peu de similitudes avec ceux d'autres instruments.

[1] Il s'agit, souvent, d'un guide d'onde.

Approche théorique

L'analyse modale des structures vibrantes est connue de longue date et s'avère particulièrement fructueuse pour l'étude des instruments de musique[1]. Initialement utilisée par les acousticiens, l'approche modale a été envisagée comme la base formelle possible d'une technique de synthèse vers la fin des années 80, par Jean-Marie Adrien [Adrien.89], Jean-Loup Florens [Florens.90] et Pirouz Djoharian [Djoharian.90].

Djoharian résume ainsi l'approche modale :

> « Le modèle modal représente la structure vibrante sous forme d'une boîte munie de deux fenêtres d'entrée et de sortie définies par une matrice et sa transposée, et contenant une série d'oscillateurs simples indépendants. (...) Les oscillateurs représentent les différents modes caractéristiques de vibration de la structure et les fenêtres représentent les déformées modales associées aux modes » [Djoharian.90][2].

Dès lors, la construction d'un modèle de synthèse modale suppose quelques étapes incontournables :

- La décomposition de l'objet à modéliser en sous-structures en interaction : membrane, tube acoustique, cloche, résonateur, chevalet, excitateur par exemple ;
- La spécification des *interactions* entre les points d'accès de ces sous-objets ;
- La caractérisation de chacun de ces sous-objets, supposés linéaires, par sa *représentation modale*.

Chaque mode d'un objet élémentaire est défini par :

- Un oscillateur du second ordre (fréquence et coefficient d'amortissement) ;
- Un ensemble de coordonnées qui représentent la forme du mode, c'est à dire son importance relative en aux points d'accès de l'objet.

La simulation du modèle complet suppose pour chaque mode de simuler un oscillateur mécanique élémentaire, et pour l'objet complet de répartir sur chacun d'eux les forces entrant par chaque « point d'accès » en effectuant le « changement de base » entre la représentation « extérieure » de l'objet et sa représentation modale.

---

[1] En physique, un mode c'est une déformée modale, une *fréquence* propre et un temps d'amortissement. L'ensemble des modes constitue une *base* sur laquelle peut se décomposer toute solution de l'équation de propagation des ondes avec ses conditions aux limites. Toute déformation d'un l'objet peut s'écrire alors comme une somme des contributions de chaque mode.

[2] Si l'on ne s'intéresse qu'à certains *points d'accès* de l'objet (extrémité d'une corde, centre d'une membrane...), les déformées modales ici matricielles peuvent être réduites à des vecteurs, chaque vecteur exprimant pour chaque mode l'importance du mode au point considéré.

Modalys

Le programme Modalys[1], initié à l'IRCAM par Jean-Marie Adrien, est aujourd'hui l'implémentation de référence de l'approche modale pour la synthèse des sons (voir par exemple [Adrien.91], [Morrison&Adrien.93]). Nous en résumerons, en conséquence, les grandes lignes.

Modalys est comme CORDIS-ANIMA à la fois un système de représentation, un langage, et un système de simulation. Il est ainsi, avec CORDIS-ANIMA, l'une des principales techniques de modélisation physique destinées aux musiciens qui « fasse système », c'est à dire qui se propose d'être le cadre conceptuel de la mise en œuvre d'un véritable processus de création du son[2].

Modalys se compose, dans sa version initiale, d'un interpréteur de commande pour la définition des modèles, d'un ensemble de primitives permettant de définir les objets élémentaires (c'est à dire les modes de ces objets) et leurs interactions et d'un moteur de simulation[3]. Modalys propose plusieurs stratégies pour la définition des modes - c'est à dire celle des objets, puisqu'un objet élémentaire s'identifie à ses modes. Il est possible de calculer les données modales d'objets homogènes et de forme simple, spécifiés par leur dimensions, leurs conditions aux limites et quelques paramètres (tension de précontrainte, densité linéique ou surfacique, module d'Young par exemple). Les modes peuvent également être spécifiés un par un. Enfin, une structure modale plus complexe peut être obtenue par hybridation des modes de deux objets.

Une attention particulière est accordée dans Modalys aux interactions entre structures. Plusieurs *connexions types* sont disponibles pour exprimer la manière dont deux objets interagissent à leur *point de contact* : collage (adhérence totale), frottement, percussion, pincement (plectre, etc. Deux objets quelconques sont susceptibles d'être connectés avec n'importe laquelle de ces interactions. Un tube acoustique peut ainsi par exemple être en relation de frottement avec une cloche.

Modalys offre également la possibilité d'appliquer à un objet en un point donné des actions unidirectionnelles sous la forme de signaux : force(t), vélocité(t), position(t) - à charge pour l'utilisateur de spécifier les signaux à utiliser.

Enfin, Modalys définit la notion de *contrôleur*. Un contrôleur est créé, sur le modèle, pour chacune des valeurs que l'on souhaite faire varier dynamiquement. Plusieurs types de contrôleurs sont proposés pour générer différentes fonctions de contrôle dynamique : constante, enveloppe, sinus. Un contrôleur *dynamique* est un contrôleur dont la courbe peut être modifiée à tout instant de la simulation ; avec lui, des signaux évènementiels (MIDI par exemple) peuvent être convertis, lors de leur arrivée, en signaux utilisables dans la simulation (de type force, vélocité ou autre). Enfin, un *contrôleur universel* permet à l'utilisateur d'implémenter ses propres fonctions de contrôle.

---

[1] Anciennement Mosaïc

[2] Notons en outre que Modalys a été recemment intégré en tant que bibliothèque pour la synthèse de sons à l'environnement de Composition Assistée par Ordinateur OpenMusic, également développé à l'IRCAM [Polfreman.02].

[3] Une interface utilisateur, Modalys-ER, est en cours de développement à l'Université du Hertfordshire. Par ailleurs, l'implémentation temps réel de Modalys (au sein de l'environnement jMax) est proposée.

Confrontation de l'approche modale aux critères

Dans l'approche modale, une démarche structurelle et fonctionnelle cohabitent. Les données modales constituent un modèle fonctionnel de chacun des sous-objets : la décomposition structurelle s'arrête donc au niveau des sous-objets mis en interaction.

En conséquence, et la *diversité phénoménologique* est relativement faible. La visualisation d'un modèle modal en cours de simulation n'est pas par exemple une fonctionnalité native de la technique. De même, l'approche modale est très orientée vers la représentation des corps oscillants et notamment des structures vibrantes *sonores*.

La qualité sonore des modèles modaux n'est plus à prouver. Cependant, l'approche modale suppose que la « matière » soit linéaire et limite les comportements non-linéaires aux seules interactions entre objets. Seuls sont représentables sous forme de modes les instruments usuels pour lesquels on peut considérer, en *première approximation*, que les différents constituants sont linéaires (c'est le cas, généralement, des résonateurs). Cette contrainte de linéarité interdit la représentation de phénomènes non-linéaires internes à la matière[1], de sorte que la *diversité des métaphores possibles* est intrinsèquement limitée – comme dans le cas des guides d'ondes.

Par ailleurs, rien n'impose que les données modales définissant un « objet » n'aient une possible interprétation physique. Il y a là un intérêt de par la possible ouverture de la technique à la synthèse d'objets modaux irréalistes, mais aussi à l'inverse un risque : l'approche modale, si elle est envisagée sans contrainte, ne présente pas une importante *robustesse de la plausibilité*.

En fait, c'est probablement le *modèle mental* qu'on peut associer à la technique qui en est l'un des avantages fondamentaux. Si elle émane bien d'une analyse physicienne, la notion de mode n'est pas étrangère au musicien au fait de l'analyse de Fourrier des signaux sonores. Dès lors, le *modèle mental* attaché à l'approche modale est sans conteste intéressant : il permet une voie intermédiaire entre modèles (additifs) de signaux et modèles physiques[2]. L'un des intérêts majeurs de l'approche réside sans doute dans cette passerelle entre modèle physique et phénomène.

Il est cependant à l'inverse responsable de certaines de ses limites. C'est que ce même modèle mental soutient mal l'idée d'un travail avec la « matière » ; dès lors qu'on a accès aux modes (ou encore aux *contrôleurs* avec Modalys), le versant *signal* de l'approche risque de prendre le pas sur son côté *physique*. Ainsi, nous considérons que l'approche modale ne permet pas de porter à son terme le paradigme de la modélisation physique, elle laisse en suspens plusieurs des mutations pourtant fondamentales que son apparition au sein de la création musicale laisse espérer.

[1] Il est possible de simuler, dans Modalys, le couplage entre les vibrations transversales et longitudinales d'une corde. Cela suppose toutefois de créer deux objets élémentaires (représentant respectivement les modes transversaux et longitudinaux), puis de les mettre en situation d'interagir par certains de leur points de contact, simulant ainsi une non-linéarité de la matière. Une telle démarche relève bien d'une approche fonctionnelle et non pas structurelle.

[2] L'approche modale apparaît ainsi comme le pendant, dans le monde des modèles physiques, de la synthèse additive.

3.6. L'approche particulaire ; CORDIS-ANIMA

Nous en venons, enfin, aux approches particulaires modulaires, dont le formalisme CORDIS-ANIMA constitue l'une des principales mises en œuvre pour la création[1].

L'approche particulaire est couramment qualifiée de *traditionnelle* dans la littérature[2], et ce alors même qu'un nombre limité de chercheurs et de musiciens l'ont mise en œuvre. C'est qu'il y a, en fait, une confusion entre l'approche traditionnelle (de Ruiz par exemple) et l'approche particulaire modulaire.

Il est vrai qu'en discrétisant un modèle continu (l'équation de propagation dans les cordes par exemple) par une technique d'analyse numérique simple (les différences finies par exemple) on obtient souvent un modèle discret qui est *équivalent* à un réseau de particules et de liaisons (en CORDIS-ANIMA, un *chapelet* constitué de MAS et de REF). Notons, incidemment, qu'une telle discrétisation introduit un biais numérique relativement important par rapport au modèle continu : l'approche particulaire ne vaut pas par sa précision (C'1).

La réciproque n'est cependant pas correcte : lorsqu'on *assemble* des modules élémentaires, on obtient bien un modèle physique dans la mesure où son comportement émerge de l'assemblage d'éléments physiques. Par contre, rien n'impose qu'un tel assemblage corresponde à un quelconque objet réel, et encore moins à un modèle continu d'un tel objet.

Dans une perspective de création, ces différences sont déterminantes : créer, c'est *agir*, c'est *faire*, et toute œuvre – ou plus simplement ici tout modèle – est nécessairement conditionnée par le processus qui a permis sa genèse.

Le processus « constructiviste » attaché à l'approche particulaire tient au choix de la *modularité* (C5), et à la teneur des modules élémentaires. Chacun des modules est certes un modèle fonctionnel d'un comportement physique, mais ce dernier est l'un des comportements les plus simples de la matière qui soit *pertinent à l'échelle des sens humains*, et ce en même temps pour l'œil, l'ouie ou la perception tactilo-kinesthésique : inertie, viscosité, élasticité, etc. Ainsi, l'approche conduit à son terme le principe de modularité au sein des modèles physiques.

L'approche particulaire ne se distingue pas par la *précision* des phénomènes engendrés[3], notamment sonores mais par sa *profondeur* (C'6). La plausibilité des sons générés est, en conséquence, particulièrement *robuste*, puisque tout modèle construit par un assemblage de modules reste un modèle physique.

De même, la *diversité et la cohérence phénoménologique* des modèles particulaires est importante. En particulier, au delà des seuls phénomènes acoustiques, les modèles particulaires peuvent être mis en œuvre pour modéliser tout phénomène dynamique apparaissant au sein de la matière, que ceux-ci soient audibles, visibles ou manipulables et perceptibles par les mains. C'est que l'approche particulaire permet la modélisation non

[1] Voir la partie I pour une description théorique du formalisme.

[2] Voir [Roads.93], [Borin&al.92] par exemple.

[3] Par exemple, si on veut simuler une corde et qu'on suppose « idéal » un modèle continu complexe (qui prendrait en compte, par exemple, la raideur et les ondes longitudinales), on trouvera sans doute des techniques de discrétisation qui seront plus « précises » (i.e. : qui introduiront un biais numérique moindre) que celles auxquelles amèneront un modèle particulaire. Il reste que pour que l'acte de modélisation soit intégré à la création, la *précision* des phénomènes générés (C'1) n'est pas le critère le plus important.

pas des *phénomènes physiques* impliqués dans la génération du son (comme la propagation par exemple) mais, plus « profondément », de la matière elle-même. L'approche n'est spécialisée ni dans une échelle de phénomènes dynamiques, ni dans une modalité sensorielle. Avec elle, il ne s'agit pas forcément de synthétiser des sons, mais de travailler un objet qui, entre autres phénomènes, génère un son, et plus de travailler la « matière » de cet objet, de la façonner, de la composer.

L'approche particulaire apparaît comme simple et naturelle. Les représentations mentales qui lui sont attachées peuvent s'appuyer sur l'expérience commune. Chaque module est expérimentable et compréhensible par tous.

L'ensemble de ces propriétés confèrent un statut particulier à l'approche. Avec elle, les modèles physiques peuvent proposer *autre chose* qu'une nouvelle approche de la synthèse.

Ce qui compte, à nos yeux, ce n'est pas tant que les modèles physiques offrent des sons plus riches ou des paramètres et des flux paramétriques plus pertinents. Ces multiples attentes relatives aux modèles physiques sont souvent présentées comme majeures, mais nous semblent être trop restreintes. Nous pensons, avec Cadoz, que « Le 'Modèle Physique' est plus qu'une nouvelle méthode de synthèse. Bien qu'essentiellement attaché à la production du phénomène sonore dans les systèmes musicaux informatiques, il annonce véritablement une mutation dans la manière même d'envisager le rôle de l'ordinateur dans le processus général de représentation et de création » [Cadoz.90b].

Il reste que les diverses approches du modèle physique ne permettent pas toutes d'opérer cette transformation avec la même efficacité.

L'approche particulaire, du fait des représentations mentales qui lui sont attachées et de sa robustesse, est en la matière particulièrement pertinente.

Avec elle, les objets (au sens général) que manipule le créateur sont alors d'un nouveau type. Faire de la musique, ce n'est plus seulement alors agencer des notes, jouer d'un instrument, composer des sons ou encore composer le son, mais c'est, justement, composer un ou des objet(s) (ici au sens d'objets physiques), ou du moins leur représentation. Au cœur du processus de création, pour utiliser un terme galvaudé mais explicite, apparaît la notion de « réalités virtuelles » : il ne s'agit pas seulement de générer des phénomènes qui donnent l'illusion du réel mais, plus profondément, de travailler, composer avec des modèles qui en eux-mêmes sont des métaphores du réel, qui, en raccourci, paraissent réels tout en ne l'étant pas.

Annexe B
Etude d'un objet CORDIS-ANIMA bidimensionnel

Dans la partie IV, nous avons résumé quelques-uns des principaux enjeux qu'adresse le choix de la spatialité des objets CORDIS-ANIMA.

Cette annexe reproduit *in-extenso* l'article *Physical Modeling Synthesis: Balance Between Realism and Computing Speed* présenté lors de la conférence DAF-X 2000 sur les effets audio-numériques [Castagne&Cadoz.00].

Cet article présente l'étude d'un *filament bidimensionnel* que nous avons réalisée. Il montre ainsi *par l'exemple* comment la spatialité de l'objet multiplie les possibilités d'expression et offre une meilleure *plausibilité* aux sons.

Partant de cette étude, il introduit la liaison topologique en X^3 et montre comment elle permet, à moindre frais, de retrouver au sein des objets topologiques certaines propriétés non-linéaire des objets spatiaux.

PHYSICAL MODELING SYNTHESIS: BALANCE BETWEEN REALISM AND COMPUTING SPEED

Nicolas CASTAGNE

ACROE - ICA
INPG, Grenoble, France
Nicolas.Castagne@imag.fr

Claude CADOZ

ACROE - ICA
INPG, Grenoble, France
Claude.Cadoz@imag.fr

ABSTRACT

Physical modeling sound synthesis methods are known to be on the one hand potentially interesting in term of sound richness and probability, but on the other hand expansive to develop and compute.

This paper is dedicated to mass-interaction methods using explicit time. By studying theoretically a specific model, we define a new basic modelization toolkit using non-linear stiffness and we show how it is possible to reach a balance between computing speed and sound realism.

1. INTRODUCTION

Physical modeling is increasingly used to synthesize sounds: it allows both interesting sound realism and new synthesis controls [1]. Nevertheless, it causes a computing speed problem no matter which modelization principles we use. To simulate a physical model and synthesize a rich audio signal, a significant number of state variables are needed. Moreover, the calculations computed at each step to define the new state are complex, and use expensive functions in term of processing time.

Therefore, if the aim is not only a precise modeling but also to create a general physical synthesis environment, computing speed has to be increased as much as possible.

There are several ways to achieve this. Algorithm optimization is needed of course, but it is useful to work on and simplify the modeling bases and modeling principles themselves. A balance between synthesized sound realism and computing speed then has to be chosen.

In this paper we emphasize explicit time mass-interaction models - and particularly their spatiality, the way they are initially built and then move in the 3D virtual space. Firstly we explain how to describe spatiality in mass-interaction models and we study algorithm complexity of 3D models. We explore then an example that shows the interest of spatiality in sound. A theoretical study of this model leads us to define a new 1D-modeling base that presents an interesting balance, and to implement it within the CORDIS-ANIMA system. The choices we carried out to allow a relevant control of the new parameter of 1D models are then explained.

2. MODELIZATION COMPONENTS

We study within the paper explicit time mass-interaction models, which modularity and genericity are of interest to define a whole physical synthesis environment.

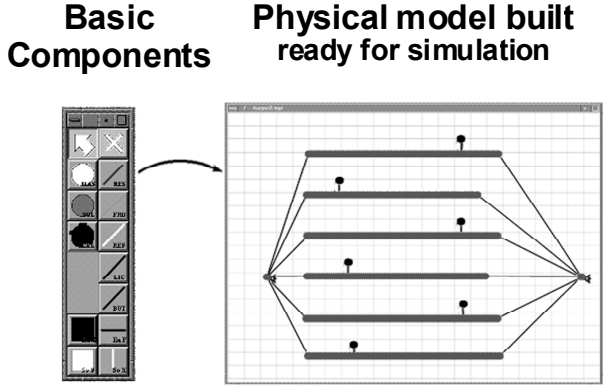


Fig 1: building a model

Such a model is built connecting basic elementary components (or "modules"), mainly masses and linear visco-elastic interactions.

Other elementary modules can be used: fixed points (infinite inertia masses), buffer interactions (effective only while the two connected masses are inside a mutual interaction field), other complex interactions implementing hysteretic behaviors…

Once parameters (physical and initial conditions) are given to every elementary module used, the model can be simulated.

Simulation uses explicit time methods, with a given frequency (44100 Hz usually). The new state (position, speed…) of each mass at the moment t=n is found computing the new forces (using state at t=n-1), then the new position.

To extract sounds from a moving model, we write in a file the position of one of the masses among one of the axis. The samples thus created can then be sent on speakers.

3. DESCRIBING SPATIALITY OF A MODEL

To describe a model, it is necessary to explain the topology of its connection network ([5], i.e. how its N masses are inter-connected through L interactions) and to give the parameters used, but also to describe its spatiality. As we shall see, choice of a spatiality is significant for sound quality and computation speed.

In order to describe efficiently a spatiality model we were led to define three spaces. Given a topology (number of masses, springs and other basic components, connection network), we distinguish the spatial appearance, the space of the movements and the simulation space.

Spatial appearance

The way a model occupies its space before simulation is called the spatial appearance.

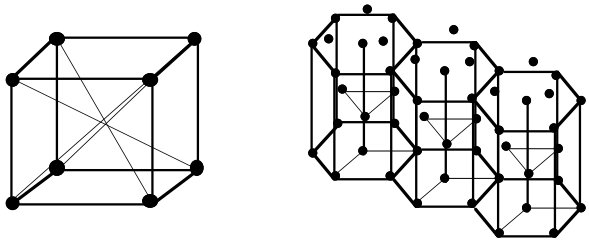


Figure 2: 3D-appearance models

Based on the spatial appearance we can define the kind of gesture (dynamic In data flow that control one of the masses' position throughout simulation) the model can receive and some of its oscillatory properties; the spatial appearance allows to distinguish wave directions, such as "transversal" and "longitudinal".

A model with a 0D-spatial appearance can be defined. In such a model, before excitation, all the masses are at the same position.

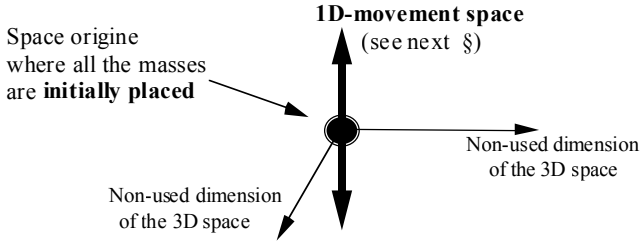


Figure 3: a 0D-appearance model…
with 1000 masses and springs !
Here transversal and longitudinal waves
cannot be distinguished.

Space of the movements

The space of the movements is the space among which the masses can move. Its dimension is proportional to the model's number of dynamic variables.

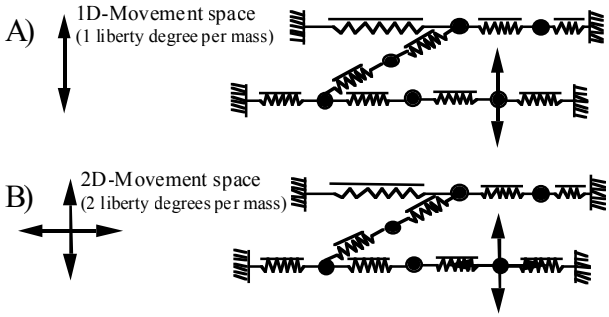


Figure 4: a 2D-qppearance model
top (A): with a 1D-movement space
bottom (B): with a 2D-movement space

In a 1D-movement model, only three variables have to be computed on each step per mass: position, speed and force. In a 2 or 3D-movement model with the same number of masses and springs, the number of calculations needed doubles or triples.

Simulation space

We call simulation space the mathematical summon of spatial appearance and movement space. It is the minimum space that has to be defined in the program in order to simulate the model.

The simulation space implies the mathematical distance used to perform the simulation.

As a consequence, the dimension of the simulation space implies the linearity or non-linearity of the model's equation. A 1D-simulation space model built with masses and springs is a linear model because the 1D distance is linear and thus the equations that define its movements are linear. 2D or 3D-simulation space models are no more linear.

Models with a complex spatiality do present, as we shall see, some sound specific interest, but needs an important computing time due to their algorithm complexity. The next paragraph studies the algorithm complexity of such spatially complex models.

4. ALGORIYHM COMPLEXITY OF 3D MODELS

The complexity study of the algorithms for a 3D-simulation space model (each mass has 3 freedom degrees) built with N masses and L spring-frictions leads the following results:

1. To obtain a rich sound timbre, N is important
1. The state system is described with at least 2*3N variables: position and delayed position of each mass on the three axes. These variables have to be calculated on every simulation step.
2. L distances are calculated for each step, so that the sqrt function is called L times - L is often greater than 1000.
3. At each step, the calculations needed are about :
   - *(30*N + 14*L) additions*
   - *(9*N + 8*L) multiplication*
   - *L calls to the sqrt function*
   - *(6*N + L) assignments*

Simulating is time expensive, mainly because of distance calculations and the freedom degrees for the masses, i.e. because of model spatiality.

5. STUDY OF A SPATIAL STRING-LIKE MODEL

Studying a specific model and listening to its sound one can show how using spatiality in physical modeling is of interest for the ear.

5.1. The experimentation model

The string-like model we used to study the sound potentiality of spatiality is made of 50 masses. Each mass has two freedom degrees and is connected to the next one with a visco-elastic

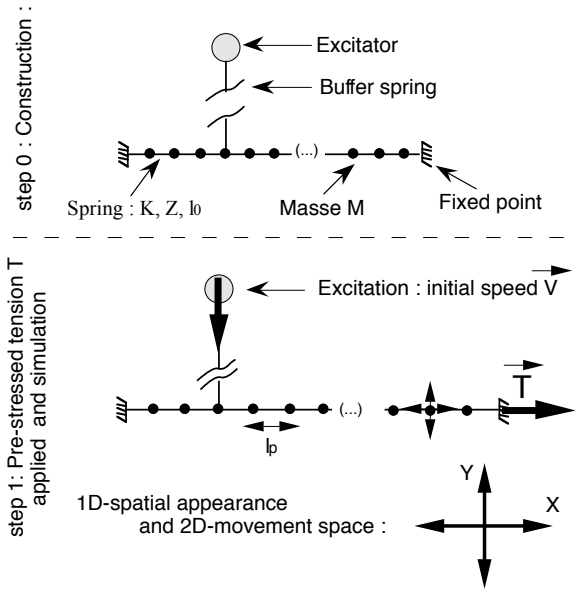


Figure 5: the 2D string-like model

interaction. The "string" is attached at its extremities and receives a pre-stressed tension. In term of spatiality, it is a 1D-spatial appearance, 2D-movement space and 2D-simulation space model.

The parameters of the model are:
- *M the inertia of each mass*
- *T the pre-stressed tension applied one its extremities*
- *Lp the length of a spring under the effect of the pre-stressed tension*
- *K qnd Z the stiffness and damping of the springs.*

In order to excite the model a percussive excitator built with a mass and a buffer visco-elastic interaction is launched on one mass of the string. The sound is made during the simulation writing on each step the position among the Z-axis of the 10th mass in a floating point sound file.

The string-like model equations are not linear, because the 2D distance calculations call the sqrt function:

$$d(mass1,mass2)=sqrt((x_1-x_2)^2+(y_1-y_2)^2)$$

and this physical non-linearity can be heard in sound.

5.2. Non-linear sound effects in the spatial model

The spatial non-linearity is not perceptible with insufficient excitation levels, but becomes hearable when excitation increases. Then, two categories of sound effects are found according to the physical parameters used (inertia, stiffness...).

First non-linear sound effect of spatiality

In small parameter fields, a modulation of the fundamental frequency around its medium value rises. This case is shown off on the next sonogram.

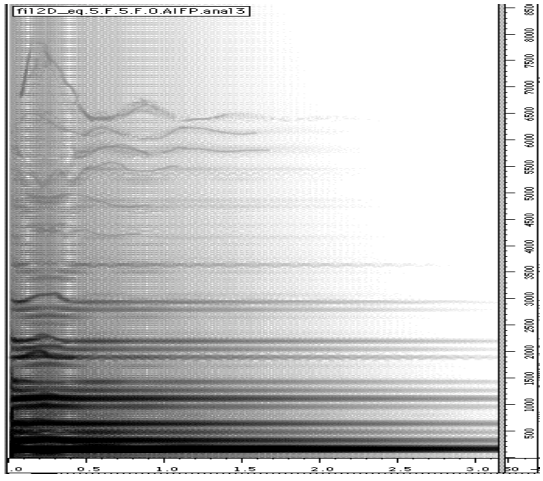


Fig 6: sonogram of a model that shows off the first non-linearity

This non-linear effect appears when the longitudinal and transversal modes are strongly coupled and when the longitudinal fundamental mode is at low frequency, i.e. in term of parameter when the springs' stiffness is small compared to masses' inertia.

The next figure is a picture of a model with such a non-linear behavior.

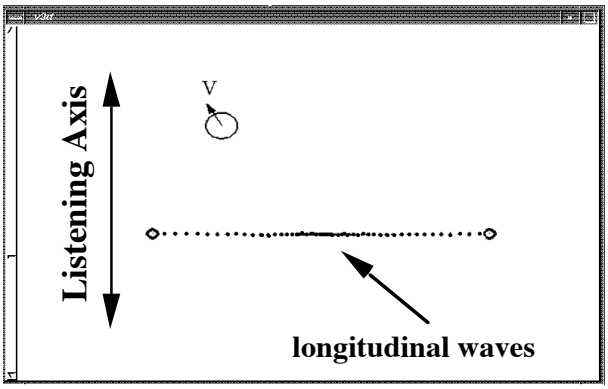


Fig 7: picture of a model that shows off the first non-linearity during simulation

In this model, the low frequency longitudinal waves modulate strongly the local transversal stiffness. Actually, this modal cooperation behavior appears in all the 2D String-like models; but it becomes hearable only for a few parameters set of values.

Second non-linear sound effect of spatiality

The second most common and most important effect is a slide of the fundamental (as shown in the next sonogram) with at the beginning, a possible hearable sound saturation

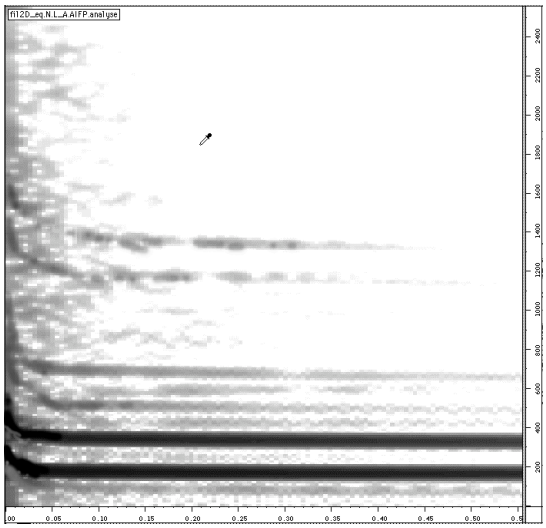


Fig 8: sonogram of a model that shows off the second non-linearity

This non-linear effect appears when the length of the string is strongly increased while it moves.

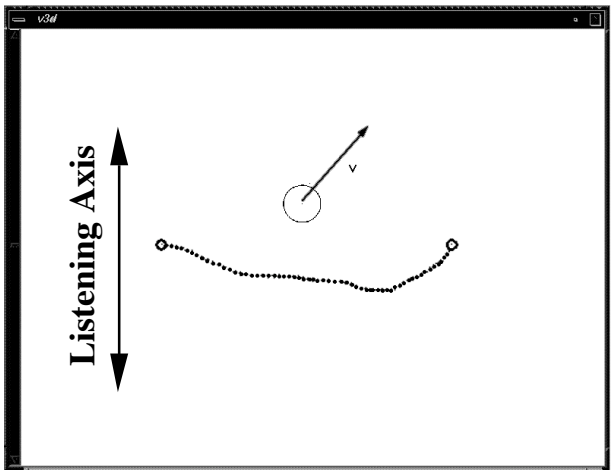


Fig 9: picture of a model that shows off the second non-linearity during simulation

This second effect is strongly correlated to the excitation level, i.e. the excitator's speed. Its physical origins are explained in many studies of membrane and string behavior [2] [3]. It is due to an increase in the stiffness forces with the string-like model lengthening, right after percussion.

This excitation-correlated sound effect is known to be important in natural sounds: the ear analyses it as the excitation level signature. Thanks to it, the ear can guess that a natural sound was made by a strongly excited object even if this sound is played or heard at a very low loudness.

5.3. theoretical simplification approach: 1D "equivalent" model

In most case when a model present the second non-linear effect the movements are mainly transversal and one can say that the longitudinal movements are mathematically negligible – which is proved both by measures and theoretical study.

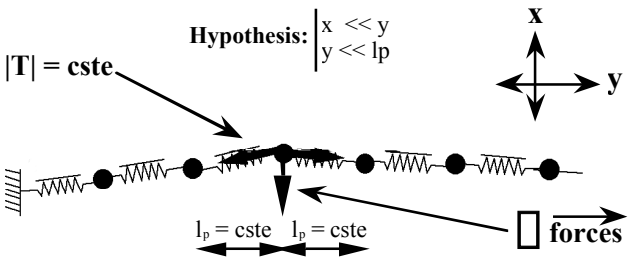


Fig 10: Simplifications hypothesis

Considering that the resulting force on a mass has no effect on movement, the force vector becomes transversal (such as the movements).

Now we can develop the transversal force using two mathematical hypotheses

2. $(x-x_0) << y$ hyp. 1 "movements are transversal"
4. $y << lp$ hyp. 2 "small deformations"

On the second order, we are led to the following results:

$$F_{n+1\to i,Y} = \alpha * DeltaY_n + \beta * DeltaY_n^3 + o((DeltaY_n)^4))$$

where

$$\alpha = -\frac{T}{l_p}$$

and

$$\beta = -\frac{K}{2 * l_p^2} + \frac{T}{2 * l_p^3}$$

where T, K, and lp are the model parameters and $deltaY_n$ is the transversal distance between the two masses at t=n.

This equation has only one dimension, and it contains:

5. a first linear term $(\alpha * deltaY_n)$, which does look like a 1D stiffness
6. a non-linear term that modulate the stiffness.

Based on it, we can try to built a new 1D-simulation space string-like model using a non-linear stiffness.

6. A 1D NON-LINEAR STRING-LIKE MODEL

According to the precedent theoretical development we defined another string-like model but with a 1D-simulation space and non-linear interaction. This model is shown on the next figure.

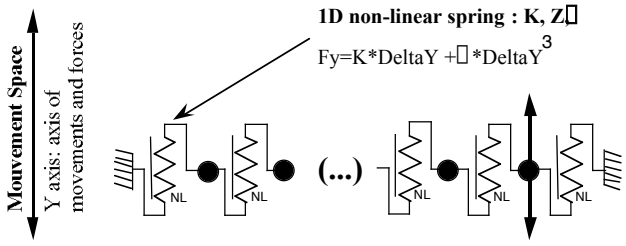


Figure 11: the 1D-simulation space equivalent model
It is a 0D-appearance model
(here 2D-viewed for representation reasons)

Experimentally, this non-linear model presents the second non-linear sound effects we found of a main interest in the 2D model, while it is height times quicker to simulate. The sound richness is a bit less than with the 2D model, but it still presents the most interesting excitation-dependant psycoacoustic characteristic.

7. A 1D NON-LINEAR ELASTICITY

The study above leads us to define a new modeling base for 1D-space movement models. Before introducing this new module, let us expose the 1D linear models' algorithm complexity.

Short study of 1D-simulation space linear models

Because the distance between two masses is, in such models, a linear function of the masses positions on the movement axis Y and because only one variable describe a mass position, a few calculations are needed for each step for a N masses and L visco-elastic interaction model:

*(N+5*L) additions*
*(N + 2*L) multiplications*
*(2*N+K) affectations*

Thus, a 1D linear model with a given (N, L) complexity is much easier to simulate than a 3D one - but the sound may be of less value.

Definition of a non-linear stiffness

For 1D models we define a new stiffness algorithm that needs two parameters K and β. In this case the force becomes:

$$F_n = K * (1 + \beta * deltaY_n^2) * deltaY_n$$

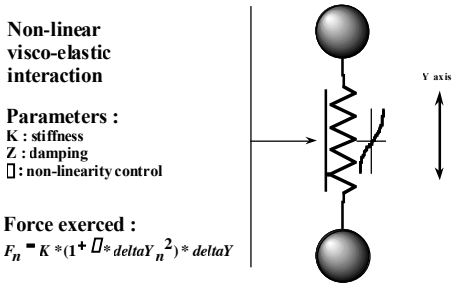


Figure 12: the new basic interaction

This stiffness force is a non-linear function of the two masses' position $deltaY_n$, with connection to the forces studied above.

To simulate a model with N masses and L non-linear visco-elastic interactions, for each step the computer needs to calculate: (N+6*L) additions

*(N + 4*L) multiplications*
*(2*N+K) affectations*

This is much less than for a 3D model with the same (N, L) complexity, and close to a simple 1D linear model.

The sounds produced by models built with the new non-linear elasticity present the second effect found in the 2D "string" sound. Timbre of non-linear 1D models depends on excitation level; the ear actually analyzes this timbre modification as a sound image of the excitation level.

8. CONTROLING THE NON-LINEAR EFFECT

Because important models cannot be real-time simulated (even though their simulation space is one-dimensional), it is of interest to draw out a process that allow a control the non-linear effect parameter.

In order to find a relevant control of this new parameter, we studied first 1D-simulation space string-like and surface-like kinds of models built with the new non-linear stiffness.

A preliminary study proved that the maximum value of $(\alpha * deltaY_n^2)$ within all the springs of the model and throughout the simulation gives a good measure of the non-linear effects that can be heard in sound.

According to this result, a relevant process was drawn in order to propose to users a control of lambda.

Two methods are possible:

The first one is a simple control of the value of lambda, which is to be given by the user.

The second supposes a linear 1D model (with linear springs) to be designed first, with its initial conditions and its gestures (and so on its medium excitation level). The process to give a relevant value to lambda is then (see also Fig. 13):

7. 1) To ask the user to give a NL coefficient between 0 and 1, describing the non-linearity he would like to perceive in the sound (NL = 0 <=> no non-linearity; NL =1 <=> important non-linearity).
8. 2) To run a first simulation on the linear model in order to calculate the maximum value DeltaYmax of $deltaY_n$ within all the springs, during the whole simulation (excitation level estimation).
9. 3) To calculate lambda as $\alpha = NL / deltaY\max^2$

Such a process allows an efficient control of the non-linear effects that will be perceived in the synthesized sound.

9. A NEW CORDIS ANIMA MODELISATION TOOLKIT: IMPLEMENTATION WITHIN GENESIS

CORDIS-ANIMA [4] is the physical-modeling language that has been developed and increased since 1975 in ACROE-ICA laboratory in Grenoble, France. Its graphically interfaced implementation, GENESIS, is a sound synthesis environment program that allow composers and researchers to define 1D mass-interaction physical models, control in different ways their parameters, and simulate them synthesizing sounds.

The 1D non-linear elasticity algorithms we drew out was added to the CORDIS-ANIMA basic component toolkit in the 1.4 GENESIS release by the beginning of September. It allows the definition of non-linear models which sound and timbre are correlated to excitation level and that are still quick to simulate. Composers will test this new basic interaction component during the next months.

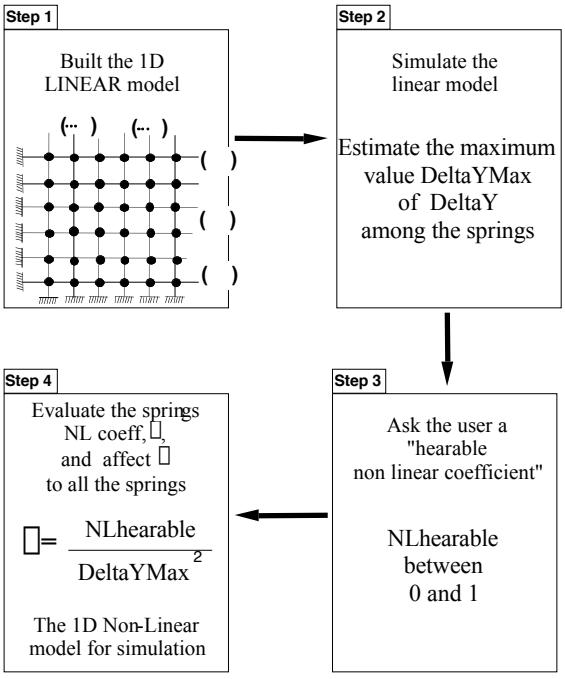


Fig 13: control of the non-linear parameter α

10. CONCLUSION

Working within the physical mass-interaction principles, we found a new interesting tradeoff between time processing and sound realism.

We added to the 1D modeling base a 1D elementary non-linear stiffness that produces the sound effect we were looking for, but at a low price in term of processing time. The natural non-linear dependence between sound level and timbre it allows, known to be important in real structures sounds, opens new ways of sound synthesis.

11. REFERENCES

- Risset, J.C., "Modèle physique et perception - Modèle physique et composition" - Colloque International Modèles Physiques Création Musicale et Ordinateur Vol.III (p711-720) - Grenoble France, 1990.
- Watzky A. , "Vibration des cordes en régime non-linéaire : modèle bidimensionnel" - *Premier congrès français d'acoustique, supplément au Journal de Physique, tome 2, 1990.*
- Cuesta, C., "Corde vibrante isolée mécaniquement, amortissements, non-linéarités ; application au clavecin et à la tempoura" - *thèse soutenue à l'université du Maine le 9 Mars 1990.*
- Cadoz, C., "Sound, Music and Image Creation with physical model - The CORDIS and GENESIS tools from ACROE", *DAFx00 Workshop, 99.*
- Incerti E, Cadoz, C "Modélisation et simulation de structures vibrantes en *CORDIS*; Matériaux et paramètres pour la création musicale" - *Journées d'Informatique Musicale, 1995.*

Annexe C
« Simuler pour connaître Connaître pour simuler » (Claude Cadoz)

Dans le chapitre 2 – partie I – et dans l'annexe A, nous avons avancé que l'approche par modèles physiques consiste à simuler les causes plutôt que les effets. Cette définition, pour générale qu'elle soit, ne satisfait pas chacun. Les termes eux mêmes, *modèle physique*, prêtent à confusion. Qu'est-ce qui est physique ? Ce qui est modélisé, ou la démarche de modélisation ? La Physique est-elle un modèle, ou le modèle est-il physique ? Ou encore : le modèle est-il l'instrument réel qu'on souhaite modéliser, ou le résultat de la modélisation ?

Il semble bien nécessaire pour savoir de quels modèles on parle de placer des jalons précis à propos de la *modélisation*, de la *physique*, de la *simulation*.

A cette fin, il nous a semblé pertinent de reproduire *in-extenso* l'article *Simuler pour connaître, Connaître pour simuler* de Claude Cadoz, publié à l'origine dans les actes du Colloque International Modèles Physiques, Création Musicale et Ordinateur, qui s'est tenu à Grenoble en 1990.

I. LA REPRÉSENTATION 395

1. De l'expérience sensible à la représentation 395
1.1. L'expérience directe du monde physique 395
1.2. Connaître - Créer ; Nécessité de la représentation 397
1.3. Conditions de la représentation 398

2. Economie de la représentation 400
2.1. Réduction de substance - Le Symbole primitif 401
2.2. Réduction d'attributs - la naissance de la Forme 402
2.3. L'abstraction 403

3. Systèmes et Langages 404
3.1. Catégorisation - Discrétisation 404
3.2. Systèmes de représentation 405
3.3. Langages - Coupure ontologique 405

II. LE MODÈLE PHYSIQUE - LA SIMULATION 408

1. Du Phénomène au Modèle 408
1.1. Expérience scientifique 408
1.2. Modèles Phénoménologique / Fonctionnel / Structurel 411
1.3. Le Modèle Physique 415

2. Le retour au Phénomène : la Simulation 415
2.1. Qu'est-ce que simuler ? 416
2.2. Simulation Phénoménologique / Fonctionnelle / Structurelle 418
2.3. Du Modèle au Phénomène : la Simulation par ordinateur 422

III. SIMULER POUR CONNAÎTRE 424

1. La simulation du physicien 425
2. Une *Physique Algorithmique* 426
3. La simulation de l'artiste 429

Simuler pour connaître / Connaître pour simuler

Réflexions sur la Représentation, la Modélisation, la Simulation
et la Création avec l'Ordinateur

Claude CADOZ

ACROE - LIFIA

Institut IMAG
46 Av. Felix Viallet
38000 GRENOBLE

Modèle Physique *est un mot passe-partout comme il s'en crée beaucoup aujourd'hui, sous la poussée de diverses nécessités qui n'ont pas toujours à voir avec celles de la science et de la connaissance. Il faut leur reconnaître une vertu toutefois : ils peuvent jouer comme appel, appel par exemple à un colloque ou plus généralement à une réflexion sur un thème spécifique. Nous en avons usé et, nous semble-t-il, avec un résultat positif.*

La nécessité de clarifier ou de fonder les concepts qu'ils pointent, ou qu'ils masquent d'ailleurs, n'en est pas secondaire pour autant. Au delà des techniques, des méthodes, des applications, il y a des enjeux. Pas seulement des enjeux directs, matériels et économiques comme on est obligé de penser automatiquement aujourd'hui quand on prononce ce terme. Des enjeux plus fondamentaux qui concernent la compréhension de l'homme par lui-même, la compréhension, au-delà des résultats (ou des faillites) matériels, par exemple de l'informatique, de ce qu'elle vient faire sur cette planète.

La création artistique tente de s'approprier l'informatique ; il n'est pas évident que là où il peut encore subsister autre chose que l'asservissement, épistémologiquement stérile et culturellement auto-décadent, de l'"offre et de la demande", ce ne soit pas l'informatique qui se soit saisi de la création artistique, avec des résultats où voisinent les illuminations pionnières et les vanités les plus affligeantes.

Il est important de laisser à certains moments le discours technique, les astuces, les prouesses et les performances technologiques un peu en retrait pour considérer, quitte à revenir à des considérations naïves, ce qui est au fond en cause.

Dans ce texte, nous voulons poser des questions, peut-être simples, mais de fond, sur la représentation, ses processus, ses conditions. Nous tentons de comprendre ce qu'est la simulation à l'aide de l'ordinateur, ce qu'elle traduit de permanent dans les méthodes et que l'on retrouve dans nos démarches et nos moyens actuels, et ce que ces derniers apportent qui n'est pas réductible au précédent.

Nous essayerons de préciser ce qu'il faut comprendre du modèle physique, ce qu'il apporte et ce qu'il ne résoud pas. Et enfin, par un parallèle entre l'artiste et le scientifique, nous essayerons de montrer, profitant du thème de la simulation, en quoi ils sont à la fois profondément identiques et profondément différents.

PREMIERE PARTIE⊡LA REPRESENTATION

1. De l'expérience sensible à la représentation

1.1. L'expérience directe du monde physique

Nous sommes au quotidien dans un monde d'êtres et d'objets avec ou à propos desquels nous pensons, agissons, percevons. Nos relations, collaborations, conflits, élaborations, transformations, mettent en jeu tous ces êtres et toutes ces choses, et au quotidien, nous ne doutons pas, à chaque respiration, qu'ils existent ou qu'ils sont ce qu'ils sont.

Or, cette évidence, support de nos états et de nos actes de tous les instants est fragile et il suffit d'une circonstance un peu particulière, d'une fièvre un peu forte par exemple pour que les choses nous paraissent différentes de ce qu'elles sont habituellement. Etre et paraître se dissocient et du même coup l'immanence et l'évidence se divisent : les êtres et les choses se dérobent et nous laissent avec notre seule conscience : conscience de ce qu'il y a nos sensations, nos actions,… notre conscience.

Notre état et nos actions jouent sur notre perception et nos sensations, les êtres et les choses sont ce que nous les faisons. Toutefois, une autre conscience nous apparaît en même temps : celle de certaines permanences de nos perceptions malgré la variation de nos actions ou inversement de variations de nos perceptions malgré la permanence de nos actions. Ainsi devons nous réviser à nouveau notre position et considérer qu'il y a d'une part nous-mêmes et notre conscience, nos moyens d'action et de perception, d'autre part l'extérieur de nous mêmes, doté d'une existence et d'une autonomie indépendantes de nous. De ce monde extérieur nous n'avons alors qu'une *connaissance*, dépendante de ce que nous sommes…

La philosophie dans son entier se pose depuis l'Antiquité les questions de la nature de la conscience, de la relation du sujet aux phénomènes, au réel, au monde, aux autres sujets, aux objets etc. On pourrait s'attarder sur les différents points de vue, les différentes *méta-physiques* et leurs évolutions depuis les pythagoriciens, Platon, Aristote, jusqu'à Kant, Hegel, Merleau-Ponty et "L'idée de la phénoménologie" husserlienne[1]… Ce serait long et difficile et pas nécessairement utile pour notre présent propos, mais il faut toutefois tenter d'expliciter les notions primitives sur lesquelles repose le développement que nous allons faire ici.

On pose qu'il y a d'une part un monde subjectif, celui qui se trouve du côté de notre conscience, d'autre part en vis-à-vis de celui-ci, un monde objectif. Nous adoptons alors, sans prétendre pouvoir sortir aussi facilement des définitions bouclées, mais en raison de ses vertus heuristiques, le point de vue général caractérisé par les quelques énoncés suivants :

- Le sujet associe à la conscience des moyens de *perception*, les sens, et des moyens d'*action*.
- Ce qui se manifeste à nos sens et ce que produisent nos actions sont des *phénomènes* et sont les *media* exclusifs entre le monde objectif et le monde subjectif. Nous admettons comme premières formes de notre perception et de notre action celles du temps et de l'espace, formes a priori de notre conscience, que nous projetons comme formes premières des phénomènes, existant donc dans le *temps* et dans *l'espace* .

---

1 Voir par exemple :
ARISTOTE, "La Métaphysique", trad. J. Tricot, Paris, 1953
G.W.F. HEGEL, "Phénoménologie de l'Esprit", trad. J. Hyppolite, Paris, 1939-1941
E. HUSSERL, "Idées directrices pour une Phénoménologie", trad. P. Ricoeur, Paris 1950
E. KANT, "Critique de la raison pure", trad. A. Trémesaygues et B. Pacaud, réed. Paris 1965
M. MERLEAU-PONTY, "Phénoménologie de la perception", Paris 1945

- Le monde objectif transcende les phénomènes : il est toujours susceptible d'être la source de phénomènes non perçus antérieurement, et sa substance se distingue a priori des phénomènes dont il est la source.

- En corollaire du dualisme sujet/objet, nous admettons la possibilité d'un pluralisme des objets, c'est-à-dire le fait que des phénomènes spécifiques puissent être associés, terme à terme à des objets spécifiques ; mais la transcendance du monde objectif par rapport au phénomène perçu interdit de considérer comme définitive toute hypothèse de multiplicité d'objets dans une situation particulière (comme dans la situation générale).

- De la même façon et pour les mêmes raisons, nous envisageons la multiplicité du sujet et l'existence d'autres consciences susceptibles d'être affectées par les phénomènes du monde objectif. Dans l'impossibilité de considérer cependant une véritable correspondance entre les états de notre conscience et ceux de la conscience des autres sujets, cette position nous conduit à envisager les autres sujets au moins comme des entités du monde objectif et par là même à nous considérer nous aussi comme partie prenante d'un tout, objectif pour d'autres sujets, et d'un tout global : le monde, ou l'univers tout court.

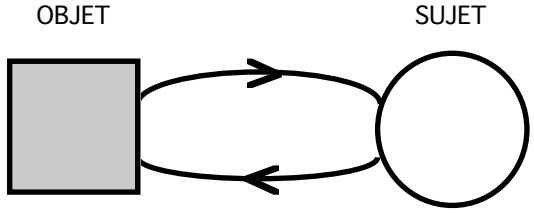
- En conséquence également de ces derniers faits et en corollaire du statut de média des phénomènes entre les objets et les sujets, nous posons la notion de phénomène comme média entre les objets eux-mêmes d'une part, et entre les sujets eux-mêmes d'autre part. Ceci conduit à considérer alors qu'il y a des phénomènes auxquels nous sommes sensibles (nous dirons *phénomènes sensibles*) ou que nous provoquons (*phénomènes émis*) et des phénomènes auxquels nous ne sommes pas sensibles ainsi que des phénomènes que nous ne pouvons pas produire. Les phénomènes que nous pouvons produire sont en général également des phénomènes sensibles ; la réciproque n'est pas vraie.

Enfin, si les objets transcendent les phénomènes, les phénomènes sont néanmoins des objets en eux-mêmes ; limites en ce qu'ils sont leurs propres phénomènes, mais objets en tant qu'ils se situent en vis-à-vis du sujet.

Objet et Connaissance phénoménologique directe

Ainsi parlerons nous d'objets, mais un objet, c'est d'une part une entité dont nous supposons la substance mais dont nous ne disposons que par sa phénoménologie : un ensemble d'actions, de perceptions et de relations entre les deux, obtenues au cours de l'*expérience sensible* (en fait sensori-motrice), et constituant la *connaissance phénoménologique directe* que nous en avons à un instant donné.

Cette position particulière de la connaissance phénoménologique par rapport à la substance ultime postulée de l'objet et de l'univers dans son entier leur confère ce que nous appellerons tout au long de ce texte un *caractère ontologique infini* ; entendons nous bien, ce ne sont pas les objets ou l'univers qui sont infinis, nous ne pouvons justement rien affirmer de définitif sur cette question, mais notre connaissance qui est finie. Il faut comprendre le mot infini comme dans la formulation astucieuse des mathématiciens : quelle que soit notre connaissance, aussi profonde soit-elle, il existe un au-delà à cette connaissance… à la petite différence près que nous devons plutôt dire : quelle que soit notre connaissance…, il *peut* exister un au-delà à cette connaissance, mais nous n'avons pas plus de certitude sur son existence que sur sa nature ni d'ailleurs que sur son inexistence.



Expérience Sensible
"phénoménologique"

1.2. Connaître - Créer ; Nécessité de la représentation

Survivre est la première préoccupation du vivant, mais si elle était la seule, il n'y aurait pas d'évolution et à la limite il n'y aurait pas de vivant car le vivant est déjà une évolution. En fait, créer, se créer est la première préoccupation de ce qui existe, c'est-à-dire de l'univers entier, même si la condition nécessaire à cela est de survivre. Il n'est pas plus illégitime de chercher une adaptation non anthropocentrique de ces mots aux organismes ou aux processus physiques les plus élémentaires, que de tenter d'expliquer le vivant, les sentiments, les aspirations humaines par les propriétés observées au niveau de ces mécanismes et processus matériels élémentaires. Il n'y a là non plus rien de téléologique car entre survivre et tendre vers un but prédéterminé, il y a un immense espace. C'est précisément celui de l'invention, de la création.

Pour créer donc, autant que pour survivre, l'homme cherche à s'adapter à son environnement, mais aussi à le et à se transformer et dans toutes les circonstances, à se servir du passé pour comprendre le présent et anticiper sur le futur. Prévoir est la première fonction de la connaissance et la première condition de la création. C'est donc selon cette fonction, pour revenir à notre propos, que l'on peut tout d'abord considérer les vertus de l'expérience directe et de la connaissance phénoménologique.

Limites de la connaissance phénoménologique directe

La réédition des mêmes perceptions, lorsqu'elle est consécutive à celle des mêmes actions conduit à l'idée de la permanence de l'objet, permanence du passé au présent. Alors apparait une possibilité de prévoir : prévoir que si l'on applique à nouveau les mêmes conduites, on obtiendra les mêmes comportements. Cette prévision repose sur un postulat fondamental : celui du prolongement de la permanence de l'objet du présent au futur.

La connaissance phénoménologique directe a donc une vertu de prévisibilité. Toutefois, cette vertu est très limitée. En effet, rien ne nous permet d'anticiper sur ce que sera le comportement de l'objet si nous envisageons par exemple de lui appliquer une conduite inédite. Pour prévoir toutes les possibilités, il nous faudrait les essayer toutes, ce qui n'est pas de la prévision mais de la vision tout court.

C'est le besoin de savoir sans devoir ou sans pouvoir essayer qui nous fait dire que nous sommes en quête de connaissances d'une autre nature.

La représentation

Prévoir ne peut pourtant se faire sans voir, la connaissance suppose la présence, aussi sommes nous dans une contradiction essentielle : il nous faut l'objet alors même que nous ne voulons ou ne pouvons l'avoir. Il nous faut sa présence en son absence. De cette contradiction, il n'est pas véritablement possible de sortir, aussi en guise de présence nous adopterons un autre objet auquel nous prêterons les mêmes attributs. Nous appelons cela le *re-présenter*. La représentation est cette démarche universelle qui lie deux entités, le représenté et le représentant, et où l'on substitue les conduites appliquées au premier par des conduites appliquées au second… pour le meilleur et pour le pire.

1.3. Conditions de la représentation

Postulation

Quelle que soit la façon dont on se procure le substitut, le processus de la représentation est soumis aux mêmes conditions : en premier lieu un lien doit être établi entre le représenté et le représentant. Ce *lien de représentation* ne peut se constituer que par la confrontation de deux expérimentations phénoménologiques directes effectives, respectivement sur l'un et sur l'autre. Ceci suppose en particulier que l'objet à représenter ne soit pas complètement et définitivement inaccessible ou absent. Mais comme par essence, l'expérimentation phénoménologique est finie, il y a en second lieu la nécessité d'une postulation : celle que le lien de représentation se prolonge au delà de ce qui est expérimenté.

Ce n'est que sous couvert de cette *postulation*, généralisation de celle que l'on a déjà rencontrée plus haut, que l'on pourra en expérimentant de nouvelles situations avec le substitut, prétendre que l'original se comporterait de la même manière.

Il est évident dès ce premier stade que la représentation absolue ne peut être, puisque si nous considérons deux objets c'est que nous savons les distinguer, donc qu'il y a d'emblée une différence dans les connaissances phénoménologiques que nous en avons. Par ailleurs, bien que nous ayons éliminé le problème insoluble de l'identité de deux objets puisqu'il ne s'agit que de confronter les expériences, les actions, les perceptions et la perception des actions, nous n'avons pas tout résolu pour autant car le problème s'est déplacé : qu'est-ce que l'identité de deux perceptions ?

Représentation mentale et représentation matérielle

En fait, dès la discussion sur la connaissance phénoménologique directe se posent déjà des problèmes implicites : un phénomène se développe dans le temps ; pour en avoir une sensation, il faut que l'on puisse rassembler en une seule unité tous les instants de ce développement, que ce qui est passé puisse être encore présent, en d'autres termes *re-présenté*. La représentation était donc déjà présente.

Qu'il s'agisse alors de nos moyens de perception, de ce qui nous permet d'unifier en un seul présent des événements ou les parties d'un événement étalés ou distincts dans le temps et dans l'espace, de les confronter, de les identifier ou de les discriminer, tout cela, qui constitue nos *facultés cognitives*, nécessairement supportées par la matérialité de notre cerveau, est déjà un monde de représentations.

Ainsi, devons nous dire que la connaissance phénoménologique suppose elle-même une représentation : une *représentation mentale.* Aux éléments de cette représentation nous commençons à savoir associer aujourd'hui des objets, des éléments et des fonctions biologiques, mais ce ne sont là que des représentations à nouveau.

Ainsi, est-ce la représentation qui est première. Mais tout semble alors tourner en rond et à nouveau nous ne savons plus où sont les choses, en nous ou à l'extérieur de nous. Le monde objectif et le monde subjectif sont comme deux miroirs face à face, capables de multiplier leur image à l'infini, et le point de départ nous fait défaut. Nous sommes en train de parler de représentations de nos représentations sans pouvoir assigner d'origine à ce cercle.

Devant ce vertige, nous n'avons plus qu'à nous accrocher solidement à un postulat primordial : celui que nous existons et que les choses existent. Ce point étant acquis, il nous reste à laisser jouer ce processus non pas comme un cercle vicieux, mais comme une spirale auto-poïétique, c'est-à-dire dans laquelle c'est précisément par le passage d'une représentation à une autre, et en particulier des représentations mentales aux représentations matérielles que s'opère le processus de création.

Pour en revenir alors à l'identification de deux perceptions, nous ne pouvons la présenter autrement, comme ce sera le cas d'un certain nombre d'autres aspects de nos facultés cognitives, qu'a priori : étant donné deux représentations mentales, nous avons le pouvoir de décréter qu'elles sont identiques, c'est-à-dire de décréter, malgré une différence nécessairement connaissable si l'on veut continuer de parler de deux entités, qu'elles peuvent valoir intégralement l'une pour l'autre. Il est assez vain de chercher qui ou quoi, à l'intérieur de nous, quelle que soit la forme de représentation que l'on puisse se donner du cerveau lui-même, détient ce pouvoir ultime de décret. Il n'y a pas de réelle solution à ce problème et prétendre qu'il s'agit d'un déterminisme régi par des lois matérielles n'en est pas une dans la mesure où ce que nous appelons *déterminisme* et *matériel* découle déjà en soit d'un certain nombre de représentations pour lesquelles nous faisons appel aux notions d'association, identification, différenciation …

Mais un tel absolu au départ n'est pas nécessaire.

Impossibilité de la représentation absolue

La seule représentation absolue d'un objet, nous l'avons déjà dit, ne peut être que cet objet lui-même puisque si nous disposons de deux objets, il existe au moins cette différence phénoménologique entre eux qui nous permet de les distinguer.

Mais il y a plus grave. La représentation contient un principe de division, c'est-à-dire, plus loin que l'idée de la multiplicité, celle de l'indépendance des objets, puisqu'elle a besoin de cette indépendance au moins entre deux objets particuliers : le représenté et le représentant. Or, rien ne nous permet non plus jamais d'affirmer, lorsque nous avons dessiné le contour d'un objet pour le mettre en représentation, que ce contour, lié à notre expérience phénoménologique, est définitif. Une expérimentation ultérieure peut nous révéler l'influence de parties jusque là ignorées.

Cette question prend d'ailleurs un sens aigu alors que la Théorie du Chaos nous apprend que des événements, des phénomènes, ou des "objets" extrêmement éloignés dans le temps ou dans l'espace, ou extrêmement infimes peuvent avoir une incidence considérable sur le déroulement de certains processus présents et locaux. C'est la fameuse image maintenant bien connue de l'"effet papillon", introduite à la suite des travaux de Lorenz, en météorologie, dans les années 60, pour évoquer l'extrême sensibilité de certains processus à de très faibles perturbations, et qui explique que le déplacement d'air provoqué par le battement d'aile d'un papillon aujourd'hui à Pékin peut se transformer en tempête le mois prochain à New York. Ce que nous avons regardé comme objet à un moment doit alors peut-être intégrer, dans un nouveau contour, d'autres éléments jugés jusque là indépendants. Inversement d'ailleurs, dans un contour déjà circonscrit, l'effet peut se manifester, le papillon peut déjà être… dans la pomme. Nous ne pouvons, compte tenu non pas de la présence systématique de tels effets, mais seulement de leur éventualité, ni vérifiable ni rejetable a priori, jamais considérer que nous avons effectué un découpage définitivement légitime. Par ailleurs, l'effet papillon a son dual : certains processus, peuvent nous interdire l'accès à ce qui fut avant ou ailleurs dans la mesure où leur état, présent et ici, est irréductiblement déterminé par des *attracteurs* absolument indépendants de ces conditions initiales ou extrinsèques.

Mais le papillon l'emporte, et son effet est double sur notre conception cartésienne du monde, il la bouleverse de deux façons : au premier ordre, lorsqu'il se manifeste effectivement et nous met réellement devant un processus chaotique ; au second ordre, en ce qu'il nous force à considérer l'éventualité permanente de son existence. Il ouvre alors une brèche définitive : tous les objets et tous les phénomènes de l'univers sont susceptibles de se révéler en interaction les uns avec les autres.

Bien que bouleversante, en nous montrant que des processus simples et déterministes peuvent demeurer imprévisibles, la Théorie du Chaos ne doit pas cependant nous faire oublier que les objets peuvent aussi résister à notre connaissance parce qu'ils sont complexes. En d'autres termes, l'effet papillon ajoute une dimension au caractère ontologique infini des objets en ce que des phénomènes extrêmement ténus, infimes, au plus profond de l'objet ou au plus loin de l'univers peuvent êtres sensibles, ici, au présent.

Quoi qu'il en soit, le résultat est que le seul objet véritablement isolable est l'univers lui-même, et qu'en fin de compte, la seule représentation absolue ne peut être que celle de l'univers dans son entier,… par lui-même.

Il nous faut tout de même représenter des objets plus petits avec des objets plus petits distincts des premiers. Pour le meilleur et pour le pire, avons nous déjà dit. Le meilleur, c'est quand la correspondance demeure a posteriori, le pire, c'est quand la divergence s'installe. Dans la mesure où nous représentons malgré l'impossibilité de la représentation absolue, tout représentant recèlera d'une manière ou d'une autre, révélée ou non, une différence avec son représenté. Des propriétés appartiendront à l'un et non à l'autre et réciproquement sans que nous puissions toujours le savoir a priori. En fait, le représentant n'est pas le représenté. Une manière d'échapper à cette triste condition est de considérer que la représentation est un processus toujours en cours, toujours en cause et à remettre en cause… et il est une tentation à combattre, la plus commune et la plus universelle depuis probablement les premières lueurs de la pensée humaine, celle de confondre abusivement le représentant et le représenté, de prendre l'image pour l'idole, d'oublier, quand on parle des propriétés et des comportements de nos représentations, qu'il ne s'agit que de celles de nos représentations…

C'est un certain parcours du processus de la représentation que nous allons essayer d'aborder maintenant.

2. Economie de la représentation

La seule représentation absolue d'un objet est cet objet lui-même. Cela ne nous avance pas beaucoup puisque nous voulons pouvoir nous passer de lui. Toutefois on peut présenter les choses autrement et considérer la représentation de l'objet par lui-même comme un point de départ, un "degré zéro" de la représentation. Sous postulation de sa permanence, c'est-à-dire de son identité avec lui-même à travers le temps et à condition que nos actes à son égard ne le détruisent ou ne le transforment pas, il est représentation de lui-même pour le futur (ainsi que pour le passé d'ailleurs).

Nous avons déjà décrit les limites de cette situation, passons sans détours aux stades suivants.

Le stade immédiatement suivant, le "degré un", c'est le premier où nous pourrions nous passer matériellement de l'original. Imaginons alors, sans nous soucier de ce qu'il faut faire pour cela, que nous soyons en présence de deux objets, nos perceptions et les perceptions de nos actions à leur égard se correspondant terme à terme. Il y a là toutes les conditions pour faire jouer à l'un le rôle de représentant de l'autre, le choix de la victime est d'ailleurs indifférent puisque selon notre expérience, ils sont "identiques". Ce choix arrêté, postulant que l'identité phénoménologique se prolongera, on pourra prédire ce qui se passerait sur l'un, en son absence cette fois, en expérimentant sur l'autre. Cependant l'intérêt reste limité car les expériences sont aussi engageantes matériellement sur le représentant que sur le représenté.

Prévoir, c'est voir en l'absence, quand la mise en présence ne peut être que future, mais c'est aussi voir, dans l'*ici* , le *maintenant* et dans les limites du champ de conscience ce qui nécessite un temps et un espace plus grands, ce qui s'étend plus tard et plus loin, et du même coup ce qui commence plus tôt. C'est demander en fait au substitut de représentation une autre propriété fondamentale et paradoxale : celle d'être identique à l'objet tout en étant moins encombrant que lui, spatialement, temporellement, celle d'impliquer une dépense, un engagement moindre de la part du sujet. De nouvelles conditions sont nécessaires, concernant aussi bien les représentations matérielles que mentales : il faut une *économie de la représentation*.

2.1. Réduction de substance - Le Symbole primitif

Ici comme précédemment, les raisonnements que l'on peut faire au sujet des substituts de représentation matériels doivent pouvoir s'appliquer aux substituts de représentation mentaux, en prenant garde, puisque le mental est aussi matériel, à ce que pour le problème des uns, on ne présuppose pas la solution dans les autres.

L'économie matérielle peut se poser d'une première manière simple : étant donné un "double" comme celui que nous avions plus haut, il faudrait, pour faire une économie, pouvoir décomposer cet objet en parties et ne considérer que l'une d'elles en lieu et place du tout. On peut simplifier encore en sautant pour l'instant la phase du double et en cherchant à décomposer l'original, supposé accessible. La question est alors : comment et dans quelles conditions ce découpage peut-il se faire et comment la partie peut-elle valoir pour le tout ? On ne peut y répondre sans recourir, précisément, au même processus que précédemment lorsqu'il s'agissait d'identifier deux objets, c'est-à-dire d'une part en mettant en jeu le vis-à-vis représentation matérielle / représentation mentale, d'autre part en faisant intervenir à la base une faculté a priori.

On est proche ici de la notion de *symbole* dans son sens étymologique grec qui dérive du verbe "joindre" et définit un objet partagé en deux, la possession de chacune des deux parties par deux individus différents leur permettant de se rejoindre et de se reconnaître. Mais dans cette métaphore, de nombreux présupposés sont implicites, et en particulier ce qui permet effectivement de déclarer que les morceaux se rejoignent suppose l'intervention d'actes et de représentations mentales à nouveau.

En fait, dans le processus de perception, il y a déjà une économie, celle qui nous permet de contenir dans le volume de notre boîte crânienne, le monde et l'espace à l'intérieur desquels elle se trouve elle-même. Cette économie procède du principe même du symbole précédent, qui s'établit tout simplement… entre *l'objet* et le *phénomène sensible*. Ramené au niveau des représentations mentales maintenant, ce principe peut s'appliquer à nouveau. C'est l'ensemble des connaissances phénoménologiques relatives à un objet qui subit un découpage. On décrète alors qu'une certaine partie vaut pour le tout ce qui signifie que c'est elle qui opère et que l'on se passe effectivement du reste, mais que l'on peut toujours, si nécessaire, effectuer le recollage et vérifier la coïncidence. Les facultés a priori sont celles de l'association, de la dissociation, de l'identification à nouveau et de la discrimination.

Ce mécanisme élémentaire est en fait extrêmement fondamental et d'un recours systématique. La vue d'un objet simple suffit parfois à évoquer des objets, des êtres, des scènes, des sentiments, des situations antérieures complexes. Cet objet fut alors probablement partie d'un tout perçu globalement dans une première expérience. La réédition des perceptions relatives à la partie qu'il constitue vaut, par "recollage" avec le tout de la perception antérieure, pour cette situation antérieure.

Il est bien évident que toutes les réserves et conditions relatives à la *postulation* se propagent ici.

Cette forme de réduction est toutefois radicale ; les deux morceaux du symbole se partagent la substance du tout. Comment peuvent-ils être représentatifs de son intégrité ? Ne peut-on trouver un substitut d'encombrement moindre, tel que l'identité et l'intégrité demeurent simultanément ? Toute la complexité du mécanisme de la représentation et de son économie est dans les stratégies pour répondre à cette question. Il faudrait… l'histoire entière de la représentation pour en donner une représentation. Nous allons dans ce qui suit, seulement donner quelques "modèles" possibles sur quelques aspects complémentaires, en espérant qu'ils auront une vertu d'"explication" un peu au delà d'eux-mêmes.

2.2. Réduction d'attributs - la naissance de la Forme

Nous ne coupons pas, si l'on peut dire, au découpage puisque pour être moins encombrant, il faut enlever des parties.

Au niveau des représentations mentales, nous avons déjà mis en avant une faculté d'identifier malgré une certaine différence. Il y a là en soi une forme de découpage, entre ce qui s'identifie et ce qui ne s'identifie pas. Ce découpage, s'il est probablement substantiel au niveau de la matérialité des supports de notre représentation mentale, ne correspond pas nécessairement à un découpage substantiel des objets extérieurs en cause. Par le jeu même de cette différence, apparait une troisième forme de découpage : entre la substance et des *attributs*. Deux phénomènes, et par leur intermédiaire deux objets, peuvent alors être différent sauf par un attribut, ou inversement identique sauf par un attribut.

Nous allons considérer respectivement, à partir de deux types d'attributs fondamentaux, deux étapes essentielles de l'économie de représentation

Réduction spatiale et temporelle

En fait, la notion d'encombrement, au noeud du problème depuis le début de ce chapitre, est une notion primitive apparaissant dès l'introduction des catégories a priori du temps et de l'espace. Parmi les attributs des objets, il y leur encombrement, leur taille temporelle et spatiale, dont nous pouvons avoir une connaissance phénoménologique. On peut dire les choses en sens inverse : notre représentation mentale possède les catégories a priori du temps et de l'espace qu'elle projette sur les objets, ce faisant, elle leur confère une taille temporelle et spatiale. Ainsi pouvons nous lier ou dissocier des phénomènes, et par leur intermédiaire des objets, différents sauf par leur taille ou, inversement identiques sauf par leur taille.

Avec la notion de taille (spatio-temporelle), peut être introduit un premier état de celle de *forme* : la forme est ce qu'il y a d'identique entre deux phénomènes dont seules les étendues spatio-temporelles diffèrent. La faculté que nous évoquons ici est alors celle d'identifier des phénomènes par leur forme, indépendamment de leur taille.

Alors, un objet, identifiable à un autre objet par la forme phénoménologique, peut être candidat au poste de représentant économique de cet objet.

Le processus de représentation comporte toujours ses phases essentielles : dans la première s'établit la confrontation purement phénoménologique respectant toutefois la différence de taille, ainsi que la permanence de cette différence. C'est l'établissement de la *similitude*. La seconde, fondée sur la postulation, inclut une évolution sensible de celle-ci : à celle du prolongement de l'identité phénoménologique au delà de l'expérience effectuée, s'ajoute celle de la validité du rapport de similitude au delà de ce qui est établi.

Une certaine coupure d'avec le monde objectif vient de s'opérer. Si dans l'univers tous les objets sont possibles, ainsi que toutes les tailles sous similitude de comportement, il est une entité qui ne peut trouver son semblable sous taille réduite, c'est l'univers lui-même. Ceci suffit pour dire que, de même que la seule représentation absolue est celle de l'univers entier par lui-même, si les phénomènes peuvent se ressembler en dépit de leur taille nous ne pouvons être garantis que par sa seule dimension, un phénomène n'a pas une singularité absolue. Par la réduction de taille, le substitut de représentation acquiert des vertus d'économie nouvelles mais dans le même temps l'existence d'un tel mode de représentation marque un premier pas vers l'abstraction : le changement d'échelle abstrait l'objet de son environnement en séparant les relations de tailles intrinsèque à l'objet des relations de taille valant de l'objet à ceux qui l'environnent ou sont susceptibles de l'environner. Et si le sujet a la faculté de concevoir une identité des phénomènes relatifs à l'objet malgré leur différence de taille, il n'en est pas nécessairement de même pour l'environnement, pour qui la différence de taille peut demeurer une condition de différence absolue.

L'objet ainsi réduit est condamné à n'interagir qu'avec le sujet ou éventuellement avec d'autres objets ayant subit la même réduction.

Homéomorphies trans-modales et spatio-temporelles

Parmi les conditions de nos représentations mentales, il y a celles qui résultent du partage opéré par nos modalités sensorielles sur ce qu'elles délimitent par ailleurs comme étant le domaine des phénomènes sensibles. Déductibles des catégories a priori du temps et de l'espace, elle introduisent dans ce partage même à nouveau une certaine nature de dualisme : si des phénomènes sont visibles, d'autres audibles, palpables etc., nous avons une faculté, malgré cette différence, d'en saisir certaines correspondances. Nous pouvons identifier des perceptions relatives à des modalités sensorielles différentes.

Plus loin, mais de manière essentiellement corrélée, une certaine correspondance peut aussi s'établir entre le développement spatial et le développement temporel des phénomènes.

La faculté d'identifier des perceptions (ou des actions) sous différence de taille se prolonge donc par celle d'effectuer cette relation entre modalités sensorielles et, plus loin, entre formes spatiales et formes temporelles : on peut identifier des formes gestuelles et des formes visuelles, des formes acoustiques, gestuelles, un mouvement visible etc, à une forme visuelle spatiale … La forme demeurant ce qui est invariant au delà de tout ce qui différent par les attributs.

L'homéomorphie trans-modale et l'homéomorphie spatio-temporelle, en tant que manifestations de nos facultés cognitives, nous permettent d'abstraire d'un degré fondamental nos représentations de leur environnement réel. Le substitut peut alors se matérialiser d'un certain nombre de façons très différentes.

2.3. L'abstraction

La réunion, pour un objet donné, de tous les substituts possibles répondant seulement au critère d'identité de forme des phénomènes malgré des différences spatio/temporelles, modale, ou de taille constitue une *abstraction* de cet objet.

L'abstraction, c'est ce qui reste lorsque l'on a considéré pour un objet donné, un certain ensemble de substituts qui lui correspondent par homéomorphie trans-modale et spatio-temporelle, quand on à tout traduit en relations et que seules les relations absolues de l'entité au monde réel sont indéfinies.

L'abstraction n'est pas un objet, ni une représentation mentale particulière mais une classe d'objets, non limitée, susceptibles d'être chacun en vis-à-vis avec une représentation mentale. La conscience de l'unité, dans l'abstraction, peut cependant conduire à penser qu'il existe une certaine unité dans les moyens matériels de la représentation mentale au delà des catégories spatiales, temporelles et modales.

L'abstraction n'est pas un objet, mais pour la manipuler, nous empruntons l'un de ses équivalents concrets, mieux, plusieurs à tour de rôle, en fonction des circonstances.

3. Systèmes et Langages

Nous nous sommes évertués jusque là à substituer un objet à un autre en considérant ces deux protagonistes comme seuls sur la scène où se joue cette représentation. Nous avons maintenant épuisé leurs ressources et nous ne pourrons aller beaucoup plus loin en restant dans une situation aussi confinée. Pour gagner, ou plutôt pour économiser d'avantage, il nous faut prendre du recul et penser que nous n'allons pas, de toute notre existence, nous contenter de représenter un objet unique et nous enfermer dans une relation idyllique et éternelle avec le substitut de notre choix.

Les objets sont multiples, les relations également et la représentation est un processus dynamique. Aussi, une forme d'économie peut être envisagée, non pas seulement intrinsèquement, dans la réalisation d'une représentation, mais d'une manière délocalisée, distribuée, dans le processus de représentation en général, dans sa dynamique.

3.1. Catégorisation - Discrétisation

Dans cette nouvelle perspective, une façon radicale d'économiser apparait : c'est d'utiliser le même représentant pour plusieurs représentés différents, et plus généralement d'utiliser un petit nombre de substituts distincts pour un grand nombre d'objets différents.

C'est la démarche essentielle de la *catégorisation*. La catégorisation consiste, au stade même de la confrontation phénoménologique initiale, à identifier au même substitut non pas un objet, mais une certaine diversité d'objets, qui fondent alors la *catégorie*. Nous n'avons besoin pour cela d'aucunes autres facultés que celles qui ont déjà été invoquées, mais il y a cependant une spécificité à la démarche de catégorisation : alors qu'une expérience peut dénoncer l'identité ou la similitude précédemment établies entre un objet et son substitut et conduire à reconsidérer la légitimité de celui-ci, dans la catégorisation, le décret d'identité, pour raison d'économie, peut être volontairement maintenu malgré le constat éventuel d'une différence.

Cette nuance mise à part, on retrouve naturellement les mêmes problèmes que dans toutes les démarches précédentes. La catégorisation a aussi son revers. Que ce soit par décret ou parce que nous ne savons faire autrement, en identifiant deux objets, nous risquons toujours de manquer la réalité qui se trouve précisément dans leur différence.

La catégorisation est, elle aussi, un procédé basique dans toute la démarche de la représentation. Nous l'introduisons ici au moment de considérer les objets dans leurs relations plutôt que dans leur nature intrinsèque. Mais avant de franchir plus définitivement ce pas, soulignons cet aspect dual de la catégorisation qu'est la *discrétisation*.

La discrétisation est l'application de la catégorisation à l'intérieur même de l'objet considéré en lui-même comme un microcosme constitué de "sous-objets" multiples. Dit autrement, catégorisation et discrétisation sont les mêmes démarches appliquées respectivement au tout en tant qu'ensemble d'objets en relation et à l'objet en tant qu'ensemble de sous-objets en relation. Dans les deux cas, l'économie réalisée se fonde sur le même principe.

En toute rigueur, pour introduire correctement la discrétisation, comme pour la catégorisation, il nous faut bien évidemment ne parler que des phénomènes, puisque la substance définitive des objets (et des sous-objets) nous échappe. Les phénomènes, dès lors qu'ils peuvent se décomposer peuvent être considérés comme ensembles de constituants entre lesquels on peut procéder, comme entre les objets d'une même catégorie. Une économie de représentation est alors possible si l'on utilise un substitut tel que les phénomènes qui lui sont associés ne sont constitués que d'un seul représentant pour chacun des éléments de décomposition que l'on peut identifier. La

discrétisation, comme la catégorisation, sont d'abord phénoménologiques, mais on a déjà vu, plus haut, de quelle manière les découpages, associations, identifications, discriminations mentales peuvent se reporter au niveau matériel.

La catégorisation et la discrétisation sont des démarches très universelles. Elles peuvent être pratiquées très consciemment, mais aussi très systématiquement par les moyens naturels et élémentaires de représentation que sont nos sens et nos systèmes perceptuels, sans que nous puissions d'ailleurs décider ou modifier quoi que ce soit.

3.2. Systèmes de représentation

Utiliser un petit nombre d'éléments pour représenter un grand nombre d'objets différents est un principe d'économie évident. Dans la mise en oeuvre de ce principe que nous venons de décrire, il y a toutefois une limite à son intérêt car une fois catégorisé, le vaste monde entre certes dans un espace raisonnable, mais il perd du même coup sa diversité. La variété des objets avec lesquels nous pouvons entrer en relation n'est ni plus ni moins que celle des catégories que nous avons constituées.

Une mise en oeuvre plus élaborée de ce principe consiste à tenter de créer des objets d'une grande variété à partir d'un petit nombre d'objets de base. Il faut alors faire entrer en ligne de compte la *structure* c'est-à-dire considérer explicitement le fait qu'un objet peut être constitué d'un ensemble de sous-objets liés entre eux. Si pour un ensemble de sous-objets donnés des relations entre eux peuvent s'établir de façons différentes alors il est possible que le nombre d'objets composites différents dépasse de loin celui des objets de base. Il faut pour cela que les objets en question et leurs relations remplissent certaines conditions. Nous n'avons pas assez d'éléments pour les préciser maintenant, mais à supposer qu'elles le soient et que les objets composites puissent être investis d'un rôle de représentation, alors, avec les objets élémentaires et leurs possibilités d'assemblage, nous disposons non plus d'un substitut de représentation, mais d'un *système de représentation*.

Il faudrait préciser ici un grand nombre de choses pour montrer dans quelles conditions un système de représentation peut se constituer. Cela alourdirait par trop notre exposé alors que nous voulons seulement pointer ici le fait qu'il y a, dans l'idée même du système de représentation, cet objectif et ce moyen d'une "amplification" réelle entre le nombre d'éléments utilisés et la variété potentielle de l'univers représenté.

L'idée du système peut s'étendre, en quelque sorte "exponentiellement". En fait, les systèmes sont imbriqués, les représentations sont des représentations de représentations, non seulement en miroir, mais poupées russes, et en combinaison des deux… Nous n'avons, dans ce déjà long parcours, qu'effleuré la surface des racines.

3.3. Langages - Coupure ontologique

Avec les systèmes de représentation puis, rétroactivement, les précédentes latitudes introduites depuis le "degré zéro" de la représentation : le symbole, la réduction de taille, l'abstraction, la catégorisation, … toutes combinables, nous disposons de moyens de créer, de nous créer des représentations ; des représentations pour notre propre compte, en tant que sujet qui parcourt le temps et l'espace, mais aussi des représentations pour *notre* propre compte en tant que sujet multiple qui parcourt l'histoire, l'évolution, la culture… Ces représentations diverses nous constituent autant qu'elles nous permettent de constituer, elles sont la mémoire autant qu'elle l'utilise : mémoire matérielle, mémoire mentale, représentations matérielles, représentations mentales…

En fait, si le dipôle mental-matériel a un sens pour le sujet individu, il en a un encore plus fort pour le dipôle individu-collectif, car l'une des fonctions de la représentation matérielle est précisément la communication, c'est-à-dire la participation à la représentation et à la création dans sa dimension foncièrement collective et historique, celle qui dépasse l'individu de deux manières : par rapport à son enveloppe corporelle et son espace propre et par rapport à sa durée de vie propre. Les représentations matérielles sont l'équivalent de la mémoire et de la représentation mentales pour le sujet humain collectif. Le sujet humain collectif est constitué de tous les sujets individuels, de leurs mémoires et représentations mentales, ainsi que de tous les objets de représentation, vecteurs entre les individus du même présent et entre les générations.

A l'échelle de l'individu, ou plutôt des individus, l'enjeu de la représentation (connaître), s'étend alors à celui de la communication (faire connaître). Réciproquement, la communication est la condition par laquelle un nouvel ordre de la représentation, de la connaissance et de la création est possible : celui qui correspond au sujet humain collectif.

Au moment de la communication, plus encore qu'à celui de la représentation pour/par le sujet isolé, l'impossibilité d'échapper aux postulations a une importance. En fait, nous aimerions à ce niveau plus qu'aux précédents, garantir qu'aucun comportement imprévu de nos objets de représentation ne viendra occulter ou se substituer insidieusement à ce que nous voulons faire connaître. Par ailleurs, dans sa réalité matérielle, la communication, comme tout processus de représentation est soumise à une condition majeure : elle est constituée d'actes qui, comme l'individu, sont localisés et temporels, ont un lieu, un début, une fin, sont frappés de finitude. Nos actes communicationnels sont de même nature que nos expériences sensibles du monde : ils passent par des phénomènes sensibles qui se développent dans des espaces et des durées finies. Ainsi, quel que soit l'objet de représentation que nous utilisons, pour communiquer avec nous mêmes ou avec les autres, même si cet objet, de fait nous transcende, possède comme nous avons dit au début un "caractère ontologique infini", la connaissance phénoménologique que nous en avons au moment présent est nécessairement finie.

Pour ces raisons, nous aimerions disposer d'objets d'un caractère spécial, tels que nous soyons certains qu'ils ne manifesteront absolument aucun autre comportement que ceux que nous connaissons. Il nous faudrait des objets "finis". Ces objets nous permettraient par ailleurs de régler une autre question : nous pourrions définitivement affranchir le substitut dans son rôle de représentation et prétendre que ce que nous n'arrivons pas à connaître (a fortiori à communiquer), ce qui nous échappe, est dans l'objet original et non dans sa représentation.

Mais comment peut-on se doter d'objets dont la finitude est garantie ? A vrai dire, il n'y a pas de solution. Ces objets n'existent pas. A nouveau, nous avons recours à un stratagème, fondé sur une utilisation à rebours de la catégorisation. Ce qui constitue l'original, le point de départ, cette fois, est non pas l'objet, transcendant notre expérience, mais l'expérience elle-même, la connaissance phénoménologique, par essence finie. Nous cherchons alors, pour cette connaissance, tous les objets possibles qui peuvent s'identifier avec cette référence et qui conviendront, selon les circonstances, à l'usage qu'on en veut faire. Si, par mésaventure, l'un de ces objets occasionnels s'avise de s'écarter du rôle assigné, révèle et met en jeu un comportement qui ne correspondant pas aux phénomènes prédéfinis, nous l'éliminons *catégoriquement* et en prenons un autre.

Ainsi croyons nous avoir inventé les "objets finis".

Langages

Le moyen le plus élaboré dont nous disposions jusque là pour constituer des représentations était le *système de représentation*. Il faut alors souligner que le problème que l'on vient d'évoquer y était présent. En effet les composants du système de représentation sont jusque là des objets du monde réel tout autant que nos premiers substituts. De ce fait, ils transmettent, comme par épidémie, leur caractère ontologique infini à l'objet composite lui-même. La "substance" de l'objet qui représente, dans cette situation, tient à la fois à sa structure, que nous pouvons maîtriser puisque nous agençons les composants, et à la nature des composants eux-

mêmes, que nous ne maîtrisons que pour ce que nous en connaissons. Ainsi, la nécessité de recourir à des "objets finis" vaut-elle également pour les composants d'un système de représentation si nous avons à son égard les dernières exigences énoncées au-dessus.

Là se trouvent la fonction, la nécessité, la définition du *langage*.

C'est le sens premier que nous donnerons ici au mot *langage* en l'attachant aux systèmes de représentation dont les composants sont *finis* (nous ne disons pas en nombre fini) ceci afin de bien les distinguer parmi les systèmes de représentation en général et parce qu'il s'agit d'une propriété réellement fondamentale.

Le langage ultime serait le langage binaire de l'informatique. Les objets élémentaires qui y sont en cause ne nous intéressent que dans la mesure où au niveau le plus abstrait, puisque personne n'a jamais vu un électron autrement qu'au travers de médiations très sophistiquées, ils se comportent seulement selon deux possibilités qui s'excluent mutuellement. Leur "substance" plus intime, là où siègent d'autres comportements possibles ne nous intéresse pas, n'existe pas, au point que si elle s'avise d'intervenir, nous avons tôt fait d'appeler cela une panne et de jeter le composant coupable. Mais les systèmes électroniques ne sont pas les premiers du genre. Nos systèmes d'écriture, depuis au moins 6000 ans sont exactement de même nature et de la lettre au mot, les composants sont bel et bien de ces "objets finis". Si une lettre est mal formée, au moment d'écrire nous la recommençons, au moment de lire nous cherchons à savoir, en nous fiant à celles qui l'environnent, de laquelle dans l'alphabet il peut bien s'agir.

Coupure ontologique

Avec un langage, au moment réel de son utilisation, le nombre effectif de ses composants est fini, ainsi que celui de leurs possibilités de combinaison. En effet, même si, dans sa définition la base présente des possibilités d'extension infinies, la constitution ou l'appropriation de ses composants pour une réalisation effective passe par un ensemble d'actes réels, donc finis. Au temps présent, toutes les bases effectives de langage sont finies. Ainsi, même si leur variété peut être très grande, les objets de représentation élaborés à l'aide d'un langage constituent un univers "borné vers le haut", c'est-à-dire de complexité finie.

Avec un langage, par définition les objets élémentaires ne peuvent révéler isolément aucun comportement intrinsèque inconnu au préalable. La construction des représentations part ainsi d'un niveau en dessous duquel rien n'existe, ou en tout cas rien n'est reconnu. Les propriétés des objets construits tiennent exclusivement à la structure des combinaisons, au jeu des interactions et à la nature, entièrement connue, des composants de base. Dans le langage binaire de l'informatique, ces derniers sont tellement élémentaires et leurs propriétés réduites à une telle simplicité (auto-référente : être l'élément "0" c'est exclusivement ne pas être l'élément "1" et réciproquement), que les propriétés des objets construits émergent exclusivement de la structure. Les objets informatiques reposent sur un "tapis de "0" et de "1"" en dessous duquel il y a le vide absolu. Les objets de représentation élaborés à l'aide d'un langage constituent un univers "borné vers le bas", ils perdent les racines profondes qui confèrent au objets réels ce que nous appelions leur "caractère ontologique infini".

Les propriétés des objets, des représentations que nous créons avec un langage, même variés au point que nous ne saurions toutes les explorer, seront néanmoins entièrement comprises, déduites, conséquences de ce qui se place entre ces deux bornes.

Ainsi, autant le langage, selon la définition très générale que nous adoptons là, est un aboutissement dans la démarche de représentation, autant il consacre une *coupure ontologique* entre le monde réel et nos représentations.

Mais nous voulons le beurre et l'argent du beurre, c'est-à-dire que nous voudrions à la fois bénéficier de cette efficacité et de cette économie de représentation/communication et à la fois y

inclure tout l'univers. Nous voulons l'infini et le fini en même temps. Il n'y a pas d'issue immédiate à cette contradiction, c'est pourquoi avons nous déjà dit, la représentation dans son ensemble, prise comme un tout, ne peut être conçue que comme un processus *en cours* ; LE processus en cours, et si nous pouvons dire où nous sommes et dans quelle direction nous souhaitons aller, l'objectif est, jusqu'à preuve du contraire à l'infini tandis que notre vitesse est limitée. Autant dire que nous en avons encore pour un bon moment.

DEUXIEME PARTIE⊡LE MODELE PHYSIQUE - LA SIMULATION

Nous allons maintenant changer d'échelle afin de pouvoir nous poster, plus loin, à un lieu charnière pour la représentation. Les physiciens, plus généralement les scientifiques, et les artistes, par exemple les musiciens, sont naturellement très engagés dans les dédales de la représentation, mais leurs objectifs et leurs parcours sont très différents ; leurs stratégies sont parfois radicalement contradictoires. Or il existe des points singuliers où, par la force des choses ils se rencontrent. L'ordinateur et la simulation constituent aujourd'hui l'un de ces points paradoxaux où coexistent, entre artistes et scientifiques, des proximités intimes et d'irréductibles distances. C'est au voisinage de cette singularité que nous souhaitons conduire la suite de ce parcours.

Nous partons du point de vue du physicien. Précisons toutefois que le physicien auquel nous pensons, même si nous n'évoquons pas explicitement ses expériences, n'est pas celui des particules élémentaires, de la mécanique quantique, de l'astro-physique,… mais, à une échelle plus humaine, le physicien du perceptible.

1. Du Phénomène au Modèle

1.1. Expérience scientifique

Instruments d'observation

La première particularité de l'approche du scientifique est qu'avant même de se donner des représentations élaborées, il cherche à élargir le champ de sa connaissance phénoménologique, à étendre son expérience directe au delà des phénomènes sensibles. Mais comment connaître le non sensible alors qu'il n'est de connaissance que par nos actes et nos perceptions ? Percevoir ce qui n'est pas perceptible, agir sur ce qui n'est pas accessible, ce ne peut être qu'en le rendant accessible et perceptible et il ne peut s'agir encore une fois que d'une… représentation. En fait, nos moyens sensoriels sont déjà des moyens de représentation, ce sont eux qui déterminent ce qui est sensible et ce qui ne l'est pas et qui produisent des représentations intermédiaires à l'usage des couches plus profondes de notre monde mental. A ces moyens de représentation innés et internes, qui pourraient être observés dans leur matérialité, le scientifique ajoute, en symétrique, des moyens, tout aussi matériels et observables eux-mêmes, des sens externes en quelque sorte, des *instruments d'observation*.

Les instruments d'observation, sont des dispositifs matériels qui réalisent "spontanément" un processus de représentation. Nous n'avons jusque là envisagé la représentation que comme processus engageant une démarche humaine. En fait, on peut invoquer un processus de représentation à propos d'un dispositif entièrement matériel et objectif. Il faut que ce dispositif réalise une association entre deux phénomènes distincts. Ce qui est "spontané", c'est alors cette

association, résultant d'une dépendance inhérente à la constitution objective du dispositif. Par contre, l'affectation à un tel dispositif (existant naturellement ou construit spécialement), de la fonction de dispositif matériel de représentation, et par exemple d'instrument d'observation, reste une décision entièrement humaine, conditionnée par la problématique fondamentale de postulation : Il faut une phase initiale où des phénomènes interdépendants, tous deux sensibles, sont décrétés respectivement représenté et représentant et il faut ensuite postuler que si le phénomène représenté disparait du champ sensible, le phénomène représentant continue d'en être dépendant de la même façon.

Mesure, Relevé phénoménologique

Partant d'une expérience phénoménologique étendue, la démarche scientifique se donne ensuite toutes les garanties pour la légitimité et l'efficacité de ses représentations. Elle a recours naturellement à l'économie, aux réductions, à l'abstraction,… mais aussi et surtout aux langages et en particulier aux mathématiques. Plus loin, elle intègre le principe même de l'évolution et de la remise en cause de ses représentations. Apparait alors sa seconde particularité : celle de vouloir disposer d'une forme préliminaire de représentation objective aussi près que possible du phénomène, qui n'hypothèque sur aucune voie de représentation ultérieure a priori mais à partir de laquelle toutes soient possibles. Cette représentation objective minimale, c'est la *mesure*.

Le fondement de la mesure est le nombre. La mesure est la substitution au phénomène, en étroite collaboration avec le processus d'observation précédent, dont l'intervention peut se situer de multiples manières, d'un ensemble de nombres. On pourrait s'en tenir à cette déclaration triviale, pour laisser ensuite parler les mathématiques et leurs diverses façons d'organiser, caractériser, décrire, transformer, mettre en relation les ensembles, de nombres en particulier… Malheureusement, la notion sur laquelle tout repose alors - le *nombre* - bien qu'extrêmement commune, n'est absolument pas immédiate. Déterminer de quelle manière elle découle des considérations générales sur la représentation n'est pas simple. Le nombre est assurément un substitut de représentation, mais il n'est pas matériel. Il est une abstraction au sens où pour exister de manière sensible, il aura besoin d'une matérialisation en des "objets supports" variés. Pour le définir dans son essence, on peut toujours recourir aux axiomes de Peano par exemple, qui posent l'ensemble des nombres entiers et du même coup leurs propriétés et possibilités d'utilisations. Mais il manque un lien avec les objets et les phénomènes qui sont devant nous. Ce lien ne peut se définir qu'en étroite corrélation avec la notion de catégorisation / discrétisation.

Il faudrait ici un développement difficile dans lequel nous n'allons pas nous aventurer. En guise d'échappatoire, nous dirons que le nombre est ce qui reste quand on a catégorisé et que l'on oublie tout des catégories sauf leur existence.

Par contre, il est important de souligner maintenant que l'ensemble des nombres constitue un système de représentation, en ce qu'avec un même nombre d'éléments, un grand nombre de phénomènes peuvent être représentés. Mais c'est surtout un *langage*, au sens que nous avons donné ici à ce mot : lors de l'association d'un phénomène à un ensemble de nombres, c'est-à-dire au moment même de la mesure, la "coupure ontologique" est consommée par ce que nous ne disposons plus du phénomène, et par conséquent de l'objet, qu'au travers d'un ensemble fini d'"objets finis".

Laissons pour l'instant ces remarques en l'état et avant de poursuivre, ajoutons cet ingrédient utile à la mesure : De même que l'on a pu introduire des dispositifs d'observation qui établissent matériellement et "spontanément", à l'usage du physicien, une représentation sensible de phénomènes qui ne le sont pas, on peut ajouter ici, pour compléter la panoplie, des dispositifs qui permettent, tout aussi matériellement et spontanément de donner des correspondances visuelles (spatiales) à des phénomènes temporels. Exploitant nos facultés d'identification trans-modales et spatio-temporelles, ils introduisent toutes ces formes de représentations annexes, graphiques et autres, satellites de la mesure, que nous désignerons rapidement sous le terme de *relevés phénoménologiques*.

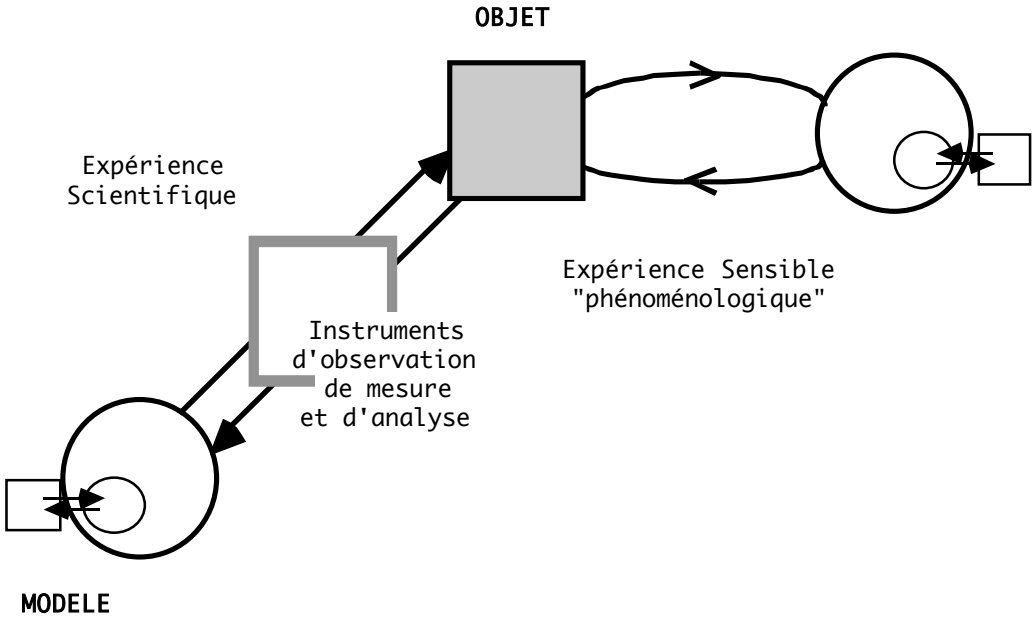
Modélisation

Dans la démarche du physicien, donc, le domaine phénoménologique direct est médiatisé par une première forme de représentation que l'on pourrait appeler *représentation phénoménologique*, dont la base fondamentale est le nombre. Une de ses principales vertus, payée au pris de la coupure ontologique est la *permanence objective*, au sens de permanence d'un objet, hors du sujet, et par conséquent sa "communicabilité" qui fait que les stades ultérieurs de représentation pourront être abordés par de multiples expérimentateurs. Toutes les représentations consécutives sont alors soumises à ce passage obligé et la confrontation sensori-motrice primitive entre l'objet et son substitut fait place à celle des représentations phénoménologiques et des mesures. En conséquence, les représentations du physicien se construisent dans l'univers du nombre, avec le langage des mathématiques.

Les mathématiques constituent un univers qui se suffit à lui-même, si toutefois on peut admettre qu'un univers peut exister, dont les objets n'ont ni forme ni couleur ni rien qui puisse se manifester à nos sens. A ceux qui sont tentés de placer l'univers des mathématiques au dessus, au delà, en deçà … de toutes choses, et en particulier comme définitivement libéré du matériel et du sensible, il convient de rappeler que sans la communication et sans le support physique de ses signes et de ses symboles, sans la poudre de craie sur la peinture noire du tableau, sans la trace de graphite ou d'encre sur la feuille… ils n'auraient jamais eu le premier soupçon de leur possibilité.

Il faudrait maintenant, pour les mathématiques, comme plus haut pour le nombre, les situer en tant que représentation. La tâche est encore plus difficile et nous n'allons pas nous risquer à réduire en quelques maladresses ce que mathématiciens, philosophes, logiciens s'attachent à développer depuis plusieurs siècles.

Pour ne pas entrer trop loin dans la complexité de la relation des Mathématiques et de la Physique, bornons nous à ce qui est essentiel à notre propos en considérant que les mathématiques sont pour le physicien un langage pour construire des représentations, langage se développant d'ailleurs souvent sous les nécessités de ce qu'il faut représenter. Ces représentations, le physicien les appelle des *modèles* ; en lieu et place de représentation, substitut de représentation, il parle de *modélisation* et de *modèle mathématique*. Ce qui est alors fondamental, c'est qu'en tant que langage, bien qu'elles sachent parler de l'infini, les mathématiques achèvent la coupure commencée avec la mesure.



1.2. Modèles Phénoménologique / Fonctionnel / Structurel

Modèle phénoménologique

La mesure donne du phénomène une représentation abstraite à l'aide d'un ensemble structuré de nombres, ensemble éventuellement très grand mais nécessairement fini. Les supports de ces abstractions peuvent être de natures très diverses, depuis les doigts de la main, origine du système décimal, jusqu'aux digits binaires (les doigts de l'ordinateur) en passant par les symboles visibles tracés sur le papier et les symboles non visibles déposés sur les matériaux magnétiques, sur le silicium etc. Bien qu'il n'y ait pas encore à proprement parler de modélisation, adoptant la même attitude qu'à propos des premiers stades de la représentation, on peut considérer le résultat de la mesure comme un "degré zéro" de la modélisation et parler alors de *modèle phénoménologique*.

Un tel modèle phénoménologique a les mêmes vertus anticipatrices que la connaissance sensori-motrice primitive : il permet simplement de prévoir, si les mêmes sollicitations (identifiées par leur mesure) sont appliquées à l'objet, sous postulat de sa permanence, que les phénomènes résultants seront les mêmes (donneront lieu aux mêmes mesures). Une authentique démarche de modélisation ne se définira toutefois qu'à partir de la recherche d'une économie et, ce qui est corrélé, de possibilités d'anticipation et de prévision effectives. La question posée maintenant est donc : partant du "modèle" purement phénoménologique, comment et sur quels principes constituer de véritables modèles ?

Modèle mathématique

Le principe d'économie peut s'exprimer d'une manière simple est générale : pour réaliser une économie à partir d'une connaissance phénoménologique donnée, il faudrait que l'on puisse ne considérer de cette connaissance qu'une partie et que celle-ci nous permette de reconstituer infailliblement la totalité. Une autre formulation peut être donnée en permutant les rôles : il faudrait qu'à partir d'une connaissance phénoménologique donnée, on puisse constituer une connaissance plus grande dont elle n'est à son tour qu'une partie. Les deux formulations sont équivalentes et illustrent simplement la correspondance entre économie et anticipation.

Dans les deux cas, il faut découvrir et mettre en oeuvre un *principe générateur*, c'est-à-dire un moyen d'engendrer un certain ensemble de nombres à partir d'un ensemble donné, de cardinal inférieur. Les mathématiques interviennent alors en tant que moyen de construire, décrire, caractériser, classer … un ensemble illimité de fonctions qui permettent précisément d'associer à des éléments d'un ensemble d'autres éléments du même ou d'un autre ensemble. L'*équation* caractérisant par exemple une loi physique est un tel principe générateur en ce qu'elle permet d'engendrer un ensemble de cardinal plus grand à partir d'un ensemble donné.

Prenons un exemple simple pour bien expliquer ce point important : nous exprimons la loi des ressorts en écrivant une équation entre la force qu'il exerce à ses extrémités et le produit de son élongation par un nombre constant appelé sa raideur. Nous pouvons alors substituer à l'ensemble des nombres qui mesurent toutes les élongations et toutes les forces qui en résultent, un ensemble plus réduit, constitué seulement de ceux par exemple qui mesurent les élongations et des symboles qui expriment cette loi. La loi en question, l'équation, permet en effet de reconstituer la totalité des données phénoménologiques initiales en associant aux valeurs de l'élongation celles de la force.

Il faut bien évidemment que les vertus modélisatrices d'un tel moyen de génération soient établies, c'est-à-dire qu'il y ait identité entre les données phénoménologiques initiales et celles qu'il produit. Les lois, opérations, fonctions, systèmes de fonctions mathématiques qui réalisent cette génération constituent en substance le *modèle mathématique*. L'ensemble des lois de la physique, les équations de la dynamique, de la mécanique, des vibrations etc… ne sont rien d'autre que des modèles et des systèmes de modèles mathématiques au sens que nous venons de donner.

Du relevé phénoménologique, ou *modèle phénoménologique*, au *modèle mathématique*, il y a justement toute l'activité de modélisation, et la question est alors, pour le physicien - l'acousticien par exemple - comment précisément passer du phénomène au modèle.

1.2.1. Analyse et Modèles Fonctionnels

De l'objet qu'il s'est mis en tête de connaître et comprendre, un superbe instrument à vent par exemple, dans lequel il a constaté qu'il y avait de fabuleux phénomènes tourbillonnaires, notre physicien a effectué une impressionnante série de mesures. Il dispose donc d'un "modèle" phénoménologique mais évidemment cela ne lui "explique" rien et il cherche un modèle plus subtil.

Plusieurs situations sont possibles.

Dans la première, il a une chance inouïe : devant lui, sur sa feuille, sans savoir ni pourquoi ni comment, est écrite une formule qui vérifie toutes les correspondances entre les conditions qu'il impose et les phénomènes qui en résultent et qui donne de ces derniers une représentation sous forme de fonction mathématique entièrement définie. Il peut alors prendre cette formule magique comme modèle et à défaut de comprendre de quelle manière elle lui a été révélée, se livrer à une série de contemplations sur les éléments qui la composent, ses propriétés mathématiques intrinsèques et prétendre que ces propriétés sont révélatrices d'une certaine essence de son objet. Il dispose d'une *physique* de son objet, d'une théorie toute faite.

Malheureusement, une telle circonstance n'est pas commune et le quotidien du chercheur de modèles est moins magique car il n'a en général, au premier abord, rien d'autre à contempler que ses données phénoménologiques. Ce qui ne manque pas de le conduire, à plus ou moins court terme, à un impérieux besoin de les mettre en pièces.

Il tente alors de décomposer le phénomène[1], c'est-à-dire d'exhiber des entités plus simples - tout en se garantissant de pouvoir reconstituer le tout par l'application d'opérateurs et de fonctions mathématiques bien définis - telles que pour celles-ci on connaisse déjà un modèle ou qu'en tout cas on puisse espérer en découvrir un plus aisément. Ce procédé correspond tout simplement ce que l'on appelle en général *l'analyse*. L'analyse, dans la démarche scientifique, fait méthologiquement suite à l'observation et à la mesure. Comme ces précédentes phases elle recourt à des moyens idéels, ici des représentations mathématiques ; on dira des *moyens conceptuels*. Mais évidemment, ces moyens se doublent ou se complètent de moyens objectifs : de la même façon qu'existent des moyens matériels d'observation existent des moyens matériels d'analyse.

L'analyse renvoie au concept de système déjà évoqué, où l'on a d'une part des éléments, d'autre part des relations ou des lois d'assemblage de ces éléments. L'analyse constitue un pas dans le processus de modélisation en ce qu'elle catégorise la démarche en deux phases : définir les lois de décomposition et modéliser les éléments de la décomposition.

Le "mur phénoménologique"

Dans le meilleur des cas alors, avec un peu de chance (mais moins que précédemment) on aura trouvé un moyen de décomposition connu nous conduisant à des éléments connus. On dispose alors d'un modèle qui s'exprime en deux niveaux : l'un relatif aux éléments, l'autre à leur combinaison.

[1] Dans cette partie du texte, nous employons pour alléger, le terme *phénomène* lorsqu'il s'agit des données numériques produites par la mesure.

Un tel modèle, quand il existe, a de nombreux avantages car il récupère au passage les vertus des systèmes de représentation, plus exactement des langages. Par ailleurs, aux propriétés mathématiques intrinsèques dont le physicien voyait, dans la formule mathématique du modèle précédent, une représentation de l'essence de son objet, s'ajoutent cette nouvelle catégorie de propriétés mathématiques : celle qui se lit dans l'expression des relations entre les éléments, dans la structure du modèle.

Malheureusement à nouveau, les choses vont rarement aussi facilement et l'on peut très bien ne trouver de modèle que pour la décomposition et non pour les éléments, ou inversement, pour arriver à des éléments modélisables, être obligé de découper arbitrairement, sans modèle. Essayant alors plusieurs façons, il se peut que nous ne parvenions pas à échapper à cette alternative. Enfin, dans le pire des cas, nous arrivera-t-il qu'aucune décomposition, même arbitraire, ne nous conduise à des modèles élémentaires connus quels qu'ils soient, peut-être parce que nous n'avons pas encore su les trouver, peut-être parce qu'il ne peut en exister dans nos mathématiques actuelles… Dans ce cas, comme dans le précédent, que ce soit en totalité ou en partie seulement, nous serons contraints de recourir à une forme de représentation *contingente*, c'est-à-dire qui ne peut s'exprimer que par un énoncé phénoménologique strict, sans règle génératrice sous-jacente, sans possibilité d'économie. Que cette description contingente apparaisse, d'ailleurs dans la caractérisation de ce qui assemble les éléments ou dans les éléments eux-mêmes ne change rien à l'affaire.

En d'autres termes, sauf pour quelques cas heureux, dans toute modélisation mathématique, il y a en général un "mur phénoménologique" infranchissable, une forme de représentation inéluctablement phénoménologique. Le modèle mathématique et le modèle phénoménologique ne peuvent être simplement opposés ; ils doivent au contraire, la plupart du temps collaborer.

Mais à nous acharner sur les phénomènes et vouloir les mettre en pièces, nous avons peut-être manqué quelque chose. N'y avait-il pas initialement deux entités ? Obnubilés par la création de ses représentations, nous avons fini par oublier l'objet.

1.2.2. Analyse et Modèles Structurels

Bien évidemment, le retour à l'expérimentation peut aider la modélisation, mais une alternative à la démarche précédente est possible si à ce niveau même, celui de l'expérimentation, on adopte une démarche analytique. Considérant l'objet lui-même comme une structure, c'est-à-dire un ensemble de sous-objets en relations, c'est lui que l'on va tenter de décomposer plutôt que le phénomène.

Analyse structurelle

Observer la structure interne d'un objet, c'est aussi une manière d'entrer en relation avec lui. Tout d'abord en considérant qu'il est constitué de parties discrètes, que ces parties peuvent être séparées effectivement sans que la décomposition se propage à notre insu pour ne nous laisser entre les mains que les rouages d'une horloge impossible à remonter. Et puis c'est considérer chacune de ces parties séparément ainsi que la façon dont elles se relient entre elles.

Remarquons que l'on peut éviter le démontage effectif. Il suffit de disposer de moyens d'observation des phénomènes relationnels entre les parties. Chaque partie peut alors être caractérisée par les comportements qu'elle manifeste à ses voisines et à l'expérimentateur en réponse aux sollicitations de celui-ci et… de ses voisines. Cependant peuvent apparaître quelques difficultés du fait que l'observateur, intrus dans cet ensemble, peut en perturber le fonctionnement sans être capable de reconnaître sa propre responsabilité. Par ailleurs, à nouveau il ne peut être certain que de tous les phénomènes, ici entre autres les phénomènes d'interaction, certains ne transcendent pas ses moyens d'observation.

Mais nous n'avons fait qu'enlever une pelure à l'oignon car devant les objets de cette décomposition les mêmes questions se posent à nouveau : en premier lieu nous n'en aurons, comme de l'objet initial, qu'une connaissance phénoménologique traduite par la mesure. Nous pourrons ensuite chercher à les modéliser, retrouvant à un moment ou un autre la même limite : l'impossibilité de caractériser les sous-objets autrement que par des données purement phénoménologiques.

Une chance supplémentaire toutefois nous est offerte car rien n'empêche de réitérer l'opération de décomposition et de pousser l'analyse structurelle d'un ordre plus loin. Mais là aussi apparait un mur car il vient nécessairement un moment où l'objet résiste, soit parce qu'il se détruit, soit parce que ses éléments deviennent inaccessibles à notre expérimentation ou encore parce que les moyens de décomposer (ou tout simplement le courage) nous manquent. A cette extrémité nous devrons alors nous contenter d'une analyse phénoménologique et finalement, probablement d'un noyau, d'un *atome* phénoménologique. Si l'on considère une fois de plus en quoi consiste toute mesure réelle, c'est à dire en un ensemble fini de données numériques, et si l'on conçoit que les relations entre les différentes représentations aux différentes échelles de cette décomposition relèvent du *langage* mathématique, on voit réapparaître ici très explicitement le fait que toute modélisation, quelle que soit la démarche adoptée, substitue à l'objet réel un objet *fini*, d'une part, et un objet partiellement *contingent* d'autre part, c'est-à-dire dont l'"explication" ultime nous échappe.

Modèle Structurel

Mais en effectuant ce dernier élément de parcours, nous venons également de mettre en évidence une notion supplémentaire. Tout d'abord, nous avons été amenés à invoquer l'idée de structure ou de système dans deux situations différentes, mettant en avant dans l'une et l'autre le fait que le dualisme élément/relation, composant/organisation constituait un paradigme fort pour la modélisation.

Le plus important est toutefois dans cet autre dualisme : la première démarche modélisatrice recourt à l'analyse, mais reste définitivement attachée au phénomène. Elle établit une relation mathématique, une *fonction*, entre les entrées et les sorties d'une "boîte noire" et si elle décompose, segmente, discrétise… elle reste résolument extérieure à la boîte noire, aussi peut-on l'appeler modélisation fonctionnelle et ses modèles des *modèles fonctionnels*. La seconde démarche recourt elle-aussi à l'analyse, mais c'est à la structure de l'objet qu'elle s'attache. Nous lui réservons ici le terme de *modélisation structurelle*.

Le modèle structurel a des points communs avec le modèle fonctionnel. Comme pour ce dernier, la connaissance à deux déterminations complémentaires, duales : l'une consiste en la caractérisation des parties, l'autre en celle de leurs relations.

Mais le modèle structurel est de loin le plus riche car les vertus "explicatives" de la représentation structurelle s'appliquent directement à l'objet. Plus loin, les propriétés de l'objet global bénéficient d'une forme d'"explication" plus élaborée qui combine les propriétés individuelles des constituants à la notion de propriété émergente, c'est-à-dire ce qui n'est dû qu'aux relations et qui ne pourrait se "voir" uniquement au niveau des éléments séparés.

Cependant aucun modèle ne peut être absolument structurel et en pratique, les trois formes de modélisation, *structurelle*, *fonctionnelle* et *phénoménologique* sont essentiellement condamnées à collaborer et à s'articuler. Le modèle phénoménologique peut être seul, ainsi que le modèle fonctionnel, par contre, le modèle structurel n'est jamais seul. Il se *ferme* nécessairement sur des modèles fonctionnels ou phénoménologiques ou les deux.

1.3. Le Modèle Physique

Venons-en au concept de modèle physique.

Il y a une certaine ironie du langage, quand il est pratiqué trop vite à des fins de justification et de rentabilité immédiates, à produire des entités ou des expressions particulièrement contradictoires ou vide de sens là où les enjeux sont les plus importants.

C'est le cas par exemple des termes *Intelligence Artificielle*, *Interface Homme/Machine* (alors que la machine n'est qu'une interface), *Réseaux Neuronaux* (quand il s'agit de réseaux d'automates cellulaires) et autres *Réalités Virtuelles*…

Le terme de Modèle Physique n'échappe pas à la règle. A son propos on rencontre toutes les dérives possibles. Dans certains cas, on emploie le mot *modèle* pour désigner les algorithmes d'ordinateur ; dans d'autres, on confond allègrement la modélisation d'un phénomène avec celle de ce qui l'engendre. Quant à la présence du mot *Physique* dans cette expression, elle n'apporte aucune information véritable car lorsque l'on modélise un signal acoustique à l'aide d'une décomposition de Fourier par exemple, ne modélise-t-on pas un phénomène physique ? Est–ce en fait ce que l'on modélise qui est physique ou est-ce la manière de le modéliser ? est-ce le modèle qui est physique ou la physique qui est modèle ?

Malheureusement, comme pour les termes précédents, nous n'avons pas le choix, et si l'on veut communiquer, il faut utiliser les termes communs. Quoi qu'il en soit, si nous employons ici le terme de Modèle Physique, que ce soit au moins avec comme définition celle que nous avons mis derrière le *modèle structurel*, appliquée aux objets physiques et avec toutes les conséquences que cela comporte à tous les niveaux.

2. Le retour au Phénomène : la Simulation

Les chemins imbriqués que nous venons de parcourir nous ont conduit l'un et l'autre de l'expérience phénoménologique directe à l'abstraction sous les espèces du modèle mathématique, fonctionnel, structurel etc… La matérialisation, le mode de manipulation et de manifestation à nos sens des modèles n'ont plus rien à voir avec ceux des objets auxquels ils se substituent ; leur nature est d'ailleurs secondaire. Par l'abstraction les objets perdent leur sensorialité ; en même temps le rôle de nos gestes et de nos sens devient simplement utilitaire : ils servent à une interrogation et à une prise d'information. Et ceci est l'aboutissement réussi d'une détermination préméditée, pour les causes de la représentation, de son économie et de son efficacité.

La *simulation* ne correspond pas à ce mouvement. Investie depuis l'ordinateur, d'une légitimité forte alors qu'elle était plutôt considérée, par les mathématiciens par exemple, comme "impure" et bâtarde, elle est un outil de plus en plus revendiqué, y compris par ces derniers. Qu'est-ce donc que la simulation ? Que vient-elle faire dans nos efforts de représentation ?

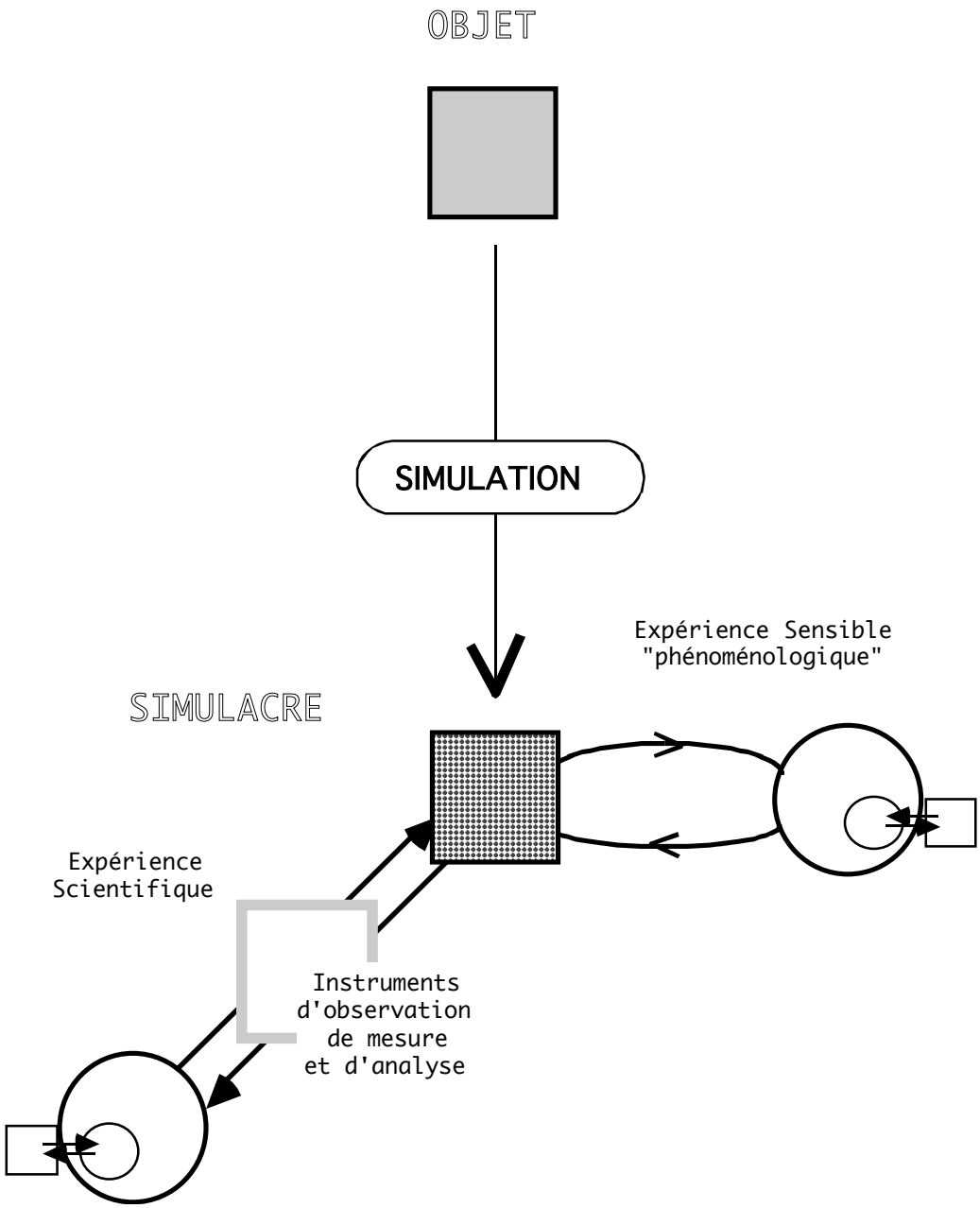
2.1. Qu'est-ce que simuler ?

Il faudrait tout d'abord bien se garder de croire que la simulation est le propre de l'ordinateur. Les automates de Vaucanson et autres géniaux inventeurs du XVIIIe siècle, sont bien, nul ne peut le contester, des simulacres, et même des "modèles" du genre. Les peintures rupestres des grottes de Lascaux sont, elles aussi, à leur manière des simulations. A l'autre extrémité, on appelle évidemment simulation les programmes d'ordinateur tels que celui qui permit à Lorenz[1] vers 1960 entre autre, de mettre en évidence le fameux "effet papillon", en débitant sur des feuilles de listings des séries interminables de chiffres représentant des phénomènes météorologiques.

Ce qui est commun à ces réalisations, au delà de leurs motivations et de leurs conditions matérielles, c'est l'importance du phénomène et de son observation. Le canard mécanique de Vaucanson n'est connu de ses spectateurs que par ses comportements visibles. Son constructeur s'est ingénié, à l'aide de mécanismes soigneusement dissimulés, à les rendre aussi proches que possible de ceux que pourrait avoir l'animal vivant. Les nombres calculés par l'ordinateur de Lorenz ne sont pas des phénomènes météorologiques tels qu'on pourrait les subir directement, mais il n'en sont pas plus loin que ce que donnerait leur mesure. Ils sont des représentations phénoménologiques. A considérer la finesse du tracé des dessins de Lascaux, on ne peut s'empêcher de penser, indépendamment de toute fonction symbolique, magique… que leur effet purement et strictement sensible (visuel) avait en soi un rôle premier, nécessaire d'ailleurs à ces autres fonctions.

Dans la simulation, il y a une même condition, une même motivation que celle qui est au départ de toute représentation : disposer, pour des raisons et des fins diverses, d'un objet qui est absent, éloigné, inaccessible parce qu'il est dans un autre espace, une autre dimension, un autre univers… et en disposer pour l'expérimenter, pour se placer en sa compagnie dans les conditions mêmes de l'expérimentation directe, sensible ou au moins phénoménologique. A quelles fins et par quels moyens, c'est ce que nous essayerons de comprendre, mais toujours est-il que la simulation se caractérise en ce que la substitution qu'elle opère, qui a bien elle aussi comme visée une certaine économie, ne se fait pas de l'objet original à un objet représentant, mais globalement de l'*expérimentation directe sur l'original* à une *expérimentation directe*… sur un *simulacre*.

[1] Voir James Gleick, "La Théorie du Chaos, vers une nouvelle science", trad. C. Jeanmougin, Albin Michel, Paris 1989.


OBJET
SIMULATION
Expérience Sensible
"phénoménologique"
SIMULACRE
Expérience
Scientifique
Instruments
d'observation
de mesure
et d'analyse

2.2. Simulation Phénoménologique / Fonctionnelle / Structurelle

Si nous pouvons décider de simuler un objet, c'est que nous en avons une certaine connaissance, au moins phénoménologique, qui nous permet de l'évoquer. Il faut donc, même si l'objet est absent ou distant, qu'une expérience minimale soit ou ait été possible. Simuler à partir d'une connaissance phénoménologique donnée, c'est alors tenter de restituer la situation expérimentale dans sa globalité, c'est-à-dire en particulier dans les mêmes conditions de cohérence, de temporalité et de dépendance entre les phénomènes relatifs à l'action et les phénomènes relatifs à la perception. Tout se présente comme au début dans la mesure où l'on doit effectuer une substitution et l'on retrouve au passage la condition de la représentation : la correspondance entre le simulacre et l'objet, établie par confrontation expérimentale, ne sera que conjecture au delà de celle-ci.

Alors, sauf à tomber par chance sur un simulacre adéquat tout prêt, nous devons engager une démarche pour le constituer. En fait, avant même de pouvoir dire quelle sorte de connaissance nouvelle nous apportera la simulation, nous devons approfondir celle que nous avons déjà. Il nous faut connaître pour simuler.

Simulation phénoménologique

Une première attitude consiste en effet, s'en prenant au phénomène (puisque c'est lui dont nous disposons), à l'analyser et le décomposer. Cherchant pour chaque élément de la décomposition des simulacres partiels, la tâche a des chances d'être plus aisée que pour le tout, globalement. On reconstituera alors le phénomène global en combinant ces derniers. Ainsi, le dessinateur, le peintre,… ceux de Lascaux comme les plus contemporains, décomposent-ils le phénomène visuel en éléments chacun reproductible à l'aide d'objets partiels, de substances visibles selon les mêmes attributs, puis réalisent une *synthèse* du phénomène en agençant librement ces objets élémentaires les uns avec les autres.

Il y a entre la mesure du physicien et cette restitution phénoménologique une différence essentielle : la seconde est sensible, la première abstraite ; mais il y a une identité profonde par le fait que les éléments qui en font la substance sont *libres* les uns par rapport aux autres. Dans les ensembles numériques qui caractérisent un relevé phénoménologique, la valeur d'un élément n'est liée à celle des autres que par le fait de leur relation de représentation au phénomène et non par leur support matériel. Dans le dessin et dans la restitution phénoménologique en général, de la même manière, les éléments supports des effets élémentaires sont libres les uns par rapport aux autres. D'ailleurs, plus la décomposition est fine et plus les objets de substitution sont élémentaires, plus grande est la liberté entre eux et, peut-on dire, plus la restitution "colle" au niveau phénoménal, à la surface de l'objet.

A côté du dessin, on peut placer l'exemple générique de la photographie (on pourrait prendre aussi la "phonographie"…). Le phénomène visuel est décomposé en parcelles : les points lumineux, dont les attributs sensibles sont portés par des objets libres les uns par rapport aux autres : les molécules colorées de l'émulsion. Toutefois, du premier au second, se produit une chose nouvelle qui est que la décomposition du phénomène, la mise en correspondance et la restitution pour chaque élément se passent de toute intervention humaine. Voisins des dispositifs d'observation et de mesure évoqués plus haut, de tels dispositifs sont des appareils d'*enregistrement et de restitution phénoménologique*.

Mais que la correspondance soit établie "manuellement" ou "automatiquement", le principe est au fond le même et il s'agit de ce que l'on pourrait appeler, par analogie avec la modélisation une *simulation phénoménologique*, degré zéro de la simulation.

Degré zéro en effet car dans cette situation, une chose essentielle est impossible : s'impliquer dans une relation expérimentale interactive, une relation dans laquelle le sujet est actif

et où les phénomènes qu'il peut observer sont dépendants de ceux qu'il produit. Ici, le sujet n'a pour toute action que le loisir de se placer en la présence du simulacre pour percevoir ses effets. Ceci n'est légitime que dans les cas particuliers où, précisément, la relation initiale avec l'objet est elle-même unilatérale : si l'objet d'origine n'est a priori accessible qu'en perception et que nos actions à son égard sont impossibles ou inutiles.

Si nous voulons une restitution de la situation phénoménologique dans ses aspects effectivement interactifs, nous rencontrons un autre ordre de difficultés. Toujours au niveau des phénomènes, il nous faut tenir compte de la dépendance entre ceux qui se rapportent à l'action et ceux qui se rapportent à ses conséquences. Pour appliquer un principe de décomposition semblable au précédent et avec les mêmes avantages, il nous faudrait pouvoir décomposer non pas chaque catégorie de phénomènes séparément, mais la totalité action/réponse en parties permettant des actions élémentaires et produisant des réponses élémentaires. Les parties seraient libres les unes par rapport aux autres, mais à l'intérieur de chacune d'elle, les actions et les réponses seraient indissociables.

En fait, une telle décomposition n'est possible que dans certains cas très particuliers où actions et réponses se correspondent exclusivement dans l'instant, ou tout au moins dans une certaine "épaisseur du présent". Plus précisément, il faut qu'à l'intérieur d'une certaine durée qui sera alors la limite de décomposition possible, les phénomènes conséquences de l'action ne dépendent en aucune manière de ce qui s'est passé en dehors de cette durée, aussi bien pour les "entrées" que pour les "sorties". Si tel est le cas, on peut alors envisager une superposition de sous-objets simulacres chacun expérimentable interactivement et agencés de façon à ce que pour chacune des actions envisageables se trouve la réponse correspondante. Les cames[1] et les systèmes à tirettes des automates du XVIIIe siècle correspondent à cette fonction. C'est aussi celle des mémoires à accès aléatoire des ordinateurs aujourd'hui.

Dans ce cas, on peut parler de synthèse phénoménologique interactive, mais malheureusement les situations que cela peut couvrir sont très singulières et relativement simplistes. On peut voir (entendre) une illustration de ce procédé, dans le monde des instruments de musique électroniques actuels, avec ces appareils qui utilisent des enregistrements de sons d'instruments réels dont une touche de clavier déclenche la lecture.

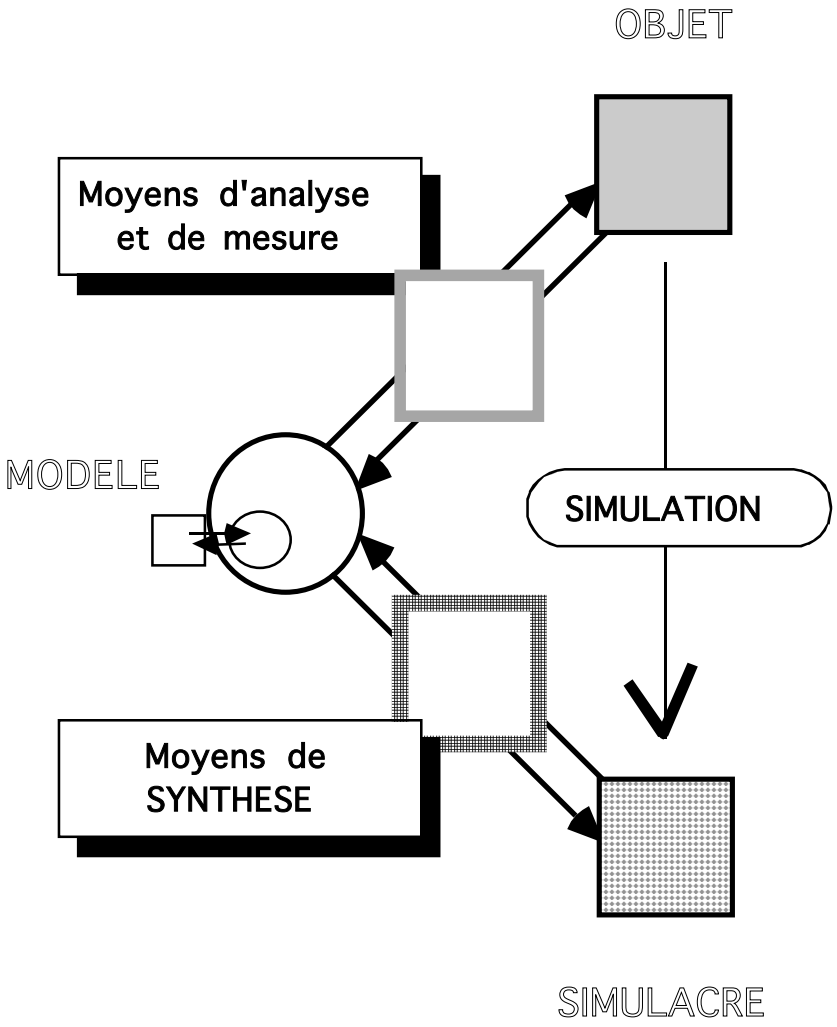
Les deux cas qui viennent d'être décrits, d'où l'on peut décliner un grand nombre de variantes et de combinaisons, restent relativement voisins dans la mesure où ils présentent l'un et l'autre des limites immédiates. Ils ne peuvent s'appliquer qu'à deux situations extrêmes, l'une où l'interaction est totalement absente, l'autre où elle est en quelque sorte *totalement présente*. Mais ils sont voisins aussi parce que l'un et l'autre, ils restent très près du phénomène. Nous proposons d'étendre au second cas le terme de *simulation phénoménologique*.

Simulation fonctionnelle, Simulation structurelle

Dans le cas général, la difficulté vient précisément de ce que les sollicitations et les réponses ne sont pas si simples dans leur dépendance temporelle : une action de courte durée peut avoir un effet étendu, une action étendue peut n'avoir d'effet qu'un certain temps après son temps propre etc. Une autre attitude est alors envisageable si on parvient en particulier à effectuer une autre forme de décomposition des données de départ : une décomposition dans laquelle chaque partie prend à sa manière en charge une correspondance entre des phénomènes de *contrôle* ou de *commande* et des phénomènes résultants et telle que, interagissant les unes sur les autres dans une recomposition globale, la correspondance soit établie globalement entre les entrées et les sorties. C'est, est-il besoin de le souligner, le pendant de l'analyse et de la modélisation fonctionnelles.

[1] Utilisée toutefois en statique c'est-à-dire par exemple en choisissant une position angulaire pour obtenir un déplacement radial, puis ultérieurement une autre, pour un autre déplacement, et en faisant abstraction du parcours qui mène de l'une à l'autre.

Il apparait maintenant clairement un parallèle entre la démarche d'analyse qui nous a conduit au modèle et celle de la … *synthèse* [2] qui nous conduit au simulacre. Ce n'est pas tout-à-fait un hasard, mais il ne faut pas manquer de remarquer ce point important : l'analyse et la synthèse vont en quelque sorte dans des sens symétriques, mais pas tout à fait cependant sur le même chemin. De chaque stade ou forme d'analyse, il faut repartir, pour simuler, vers le niveau des phénomènes, et les moyens pour l'aller ne sont pas toujours utilisables à l'inverse pour le retour.



Ici, en l'occurrence, disposant d'une analyse fonctionnelle de l'objet initial, nous avons des modèles abstraits dans lesquels apparaissent des composants dont il faudrait trouver des équivalents concrets. Pour chacun des composants, le problème se pose alors comme au premier niveau : soit nous disposons par chance d'un élément qui convient et que nous mettons alors à contribution sans plus de questions, soit il n'en est rien et nous avons le choix entre lui trouver un équivalent phénoménologique ou poursuivre l'analyse fonctionnelle à son propos.

[2] Incidemment, le terme de *simulation*, quitte à lui donner un sens un peu élargi, est à notre sens celui qui conviendrait à ce que l'on a l'habitude d'appeler *synthèse* dans les domaines du son et de l'image en particulier. Il faudrait ensuite lui adjoindre un qualificatif spécifique et parler de simulation *acoustique*, *mécanique*, *numérique*, *par modèle physique* etc.

Pour chaque sous-niveau, le problème se pose ainsi de manière récurrente, au point que nous allons, si nous n'y prenons garde, nous faire prendre dans un piège fractal. Supposons alors, pour stopper le processus, que nous trouvons un dispositif adéquat pour chacune des fonctions relevant de l'analyse fonctionnelle. Les recomposant dans un tout structuré, nous pourrons alors parler de *simulation (synthèse) fonctionnelle*. La simulation fonctionnelle se caractérise par le fait qu'elle recourt à des composants agencés et interagissant entre eux selon une certaine structure mais la nature de cette structure n'a a priori aucun rapport avec celle de l'objet ; elle résulte d'une analyse fonctionnelle des phénomènes et d'une modélisation fonctionnelle du processus de leur engendrement.

Pour revenir aux composants donc, de trois choses l'une :

- soit nous avons la chance de disposer d'un élément adéquat que nous utilisons tel quel sans nous préoccuper de ce qu'il contient - sans garanties sur sa finitude et sa stabilité, opaque à toute connaissance -
- soit il est à nouveau une structure aux composants de laquelle il faudra appliquer récursivement le raisonnement présent,
- soit enfin il est un simulacre purement phénoménologique, interactif ou non.

Les automates de Vaucanson interagissaient avec leur milieu et manifestaient des comportements différenciés. Cependant, les mécanismes internes desquels ils étaient constitués n'avaient en eux-mêmes rien de ressemblant avec l'intérieur des êtres correspondants. Construisant ces androïdes, les ingénieux inventeurs de cette époque, pour qui la technologie contemporaine (aussi prestigieuse alors que l'ordinateur l'est pour nous) était la mécanique, pensaient démontrer la "mécanicité" du vivant en faisant en sorte que tout se passe, vu de l'extérieur, conformément aux phénomènes et aux interactions connus. Pour cela, il entraient dans une simulation qui, à leurs yeux et en l'absence de possibilités d'analyses et de décomposition plus poussées, pouvait passer pour relativement structurelle (selon les termes que nous employons ici), alors que nous la prétendrions aujourd'hui plutôt fonctionnelle.

*

On aura compris alors que la troisième voie est celle que nous avons déjà adoptée pour modéliser l'objet, c'est-à-dire nous détourner provisoirement du phénomène pour nous intéresser à l'objet et nous livrer à une analyse structurelle de celui-ci, dans l'espoir de trouver, à défaut d'un simulacre tout fait pour le tout, des simulacres pour ses composants. Nous construirons alors un simulacre pour le tout, par *synthèse*, en composant les simulacres élémentaires.

La synthèse, prise dans un sens large, est une reconstruction à partir des plus petits éléments (constituant une base) pour aller vers la réalisation d'une fonction macroscopique déterminée. La démarche est donc globalement symétrique de celle de l'analyse, mais la différence fondamentale réside dans le fait que, dans le cas de l'analyse, c'est l'objet initial (la référence) qui est imposé, dans celui de la synthèse, ce sont les éléments et les règles de construction. Si les éléments et les règles de construction de base sont par leur fonction individuelles analogues à celles qui résulteraient d'une analyse structurelle de l'objet à réaliser, alors, la démarche est directe. Il suffit en effet de reproduire avec ce système, à l'identique, la construction de l'objet, composants et liaisons. L'isomorphie structurelle entre la référence et l'objet reconstruit étant garantie, le principe de causalité prend toute son "effet " : les causes étant identiques, les effets le seront aussi.

Si elle est possible, cette démarche est celle qui garantira a priori la plus grande expérimentabilité du simulacre. Parallèlement à l'analyse structurelle de l'objet, vient alors le principe d'une *synthèse ou simulation structurelle.*

Mais ça ne finit pas comme cela !

Le "mur phénoménologique" à nouveau

Pour la simulation structurelle se pose le même problème que pour la simulation fonctionnelle. Peu importe que les composants d'ordre inférieur soient des éléments fonctionnels ou des constituants effectifs de l'objet. Au moment de leur trouver un représentant concret le problème se pose de la même façon : soit nous disposons d'un élément adéquat etc., soit nous progressons d'un nouvel ordre dans la décomposition pour retrouver plus loin les mêmes questions, soit nous nous arrêtons et nous contentons d'un simulacre phénoménologique.

Dans le sens de l'analyse, nous finissions par devoir nous arrêter à une forme fonctionnelle et enfin à une forme purement phénoménologique. Dans celui de la synthèse, fixés sur une synthèse structurelle, nous ne pourrons partir en amont tout d'abord d'un certain "mur fonctionnel", puis, en amont de celui-ci d'un certain "mur phénoménologique". Aucune synthèse, aucune simulation ne peut être complètement structurelle (tout d'abord parce qu'aucune analyse ne peut l'être).

Ainsi, dans toute simulation, aussi fine soit-elle, y-a-t-il, à ses fondements nécessairement, et peut-être dans ses étages supérieurs, cohabitant avec des éléments structurels et des éléments fonctionnels :

- soit une base inconnue si l'on a recouru à des "emprunts à la nature", avec ses risques d'instabilité,
- soit une base phénoménologique fondée sur l'enregistrement pur et simple d'événements ou de correspondances terme à terme entre des événements d'entrée et des événements de sortie, sur une mémoire phénoménologique.

La substance complète de la simulation est alors une combinaison structurée, fonctionnellement ou de manière homéomorphe à la structure de l'objet initial, de ces constituants de deux natures opposées. La seule chose qui puisse être identique absolument, côté objet et côté simulacre, est, entre deux niveaux déterminés ou plus subtilement distribuée entre le plus bas et le plus haut, l'organisation structurelle. *Seule la structure des choses est véritablement imitable*.

De sorte que, au passage, il faut là aussi s'interroger sur la finitude ou l'ouverture ontologique.

Pensons, si nous avons souri plus haut de la naïveté des mécaniciens du XVIIIe siècle, que la moquerie s'adresse à nous qui croyons créer l'Intelligence et les Réalités Artificielles alors que nous ne sommes qu'un petit peu plus profond dans la structure et que ce que nous en savons est bien souvent plus fonctionnel que structurel, quand ce n'est pas purement phénoménologique.

2.3. Du Modèle au Phénomène : la Simulation par ordinateur

Venons-en maintenant à l'ordinateur dont on aura remarqué qu'à quelques allusions près, il n'était absolument pas nécessaire dans les explications qui précèdent. Il ne nous reste en fait que très peu de choses à dire pour caractériser la simulation par ordinateur. Cela ne signifie pas que ce soient les moindres.

Tout d'abord, l'ordinateur n'est pas nécessairement exclusivement une machine de simulation de la pensée ou du raisonnement. C'est avant toute chose une machine de représentation. A ce titre, elle peut prendre comme cible tout ce que l'on peut a priori identifier dans le monde réel, celui de la pensée en fait partie, comme celui des objets… disons physiques. Nous nous y intéressons ici pour le second cas de figure pour lequel nous pourrons nous passer absolument de toute référence au raisonnement, à la logique, à la déduction etc.

Qu'est alors un ordinateur en dehors de ces catégories ?

L'ordinateur effectue premièrement une mise en correspondance de deux mondes phénoménaux, celui de notre environnement et celui de phénomènes électroniques, inaccessibles à nos sens (sauf lorsqu'il s'agit d'assez hautes tensions) et à nos actions directes. Il y a, à la périphérie

d'un ordinateur tout d'abord ce que l'on doit appeler en termes techniques des transducteurs, entités à double face (ou vis-à-vis entre deux faces, cela dépend du point de vue) dont la première fonction est de donner un équivalent électronique des phénomènes du monde réel (humain et physique), dans un sens, un équivalent physique ou sensible de phénomènes électroniques dans l'autre.

Il s'agit d'ores et déjà d'une représentation. Elle a d'ailleurs un caractère spécifique très important : entre les phénomènes physiques (humains ou non) et les phénomènes électroniques qui leur sont associés, une très grande différence énergétique s'établit qui répond au critère d'économie poussé à un certain extrême. C'est l'intérêt fondamental des phénomènes électroniques qui peuvent s'identifier à des formes temporelles quelconques avec un coût très faible en énergie. Cette correspondance au niveau purement de la forme qui débarrasse les phénomènes de leur lourdeur matérielle et énergétique donne naissance au concept d'information : représentation de la forme à énergie minimale[1]. Les transducteurs, fondés sur cette catégorisation fondamentale, née au XXe siècle, entre énergie et information, l'appliquent aux phénomènes physiques pour n'en retenir que la seconde lorsqu'ils interviennent dans le sens monde réel - ordinateur, pour restituer la première (fonction *effecteur*) lorsqu'ils interviennent dans l'autre sens.

L'ordinateur est en second lieu un fantastique point de rencontre entre le modèle mathématique et le simulacre.

Entre les nombres qui ont permi la mesure, la modélisation mathématique… et les entités électroniques que nous appelons les digits binaires (les bits plus simplement), en lesquels se réduisent après un second stade de représentation, les signaux électroniques précédents, il y a une très grande proximité. Proximité parce que les bits comme les nombres sont une "essence de catégorisation", mais proximité seulement parce qu'il y a précisément une différence fondamentale : là où les mathématiques utilisent en dernier recours l'homme pour réaliser effectivement l'association entre des ensembles de nombres et d'autres ensembles de nombres, la correspondance entre des ensembles de bits et d'autres ensembles de bits se fait de manière matérielle (on dit en général, pour éluder les vraies questions : de manière *automatique*) parce que les bits en question sont et demeurent, malgré la manière formelle et abstraite de les évoquer, des objets, de différentes natures certes, mais interagissant matériellement entre eux, et c'est leur capacité d'interaction qui est la chose importante. *Ce qui est en représentation à la base, dans l'ordinateur, n'est pas la grandeur, mais l'interaction*.

C'est la simplification extrême et la maîtrise des interactions élémentaires entre ces objets qui constituent la base des "machines électroniques". Il ne faut pas dire qu'avec l'ordinateur on peut décrire en termes de logique et d'algorithmes les équations mathématiques et que la "machine les résoud alors automatiquement", nous libérant d'une tâche fastidieuse ou trop difficile, mais que les objets de base *sont les équations* elles mêmes.

Dans l'ordinateur, le concret et l'abstrait se rencontrent, là où on s'y attendait le moins, au niveau de la plus petite décomposition matérielle que l'on sache opérer et maîtriser. Et la simulation avec l'ordinateur, au lieu de reconstruire la matière des objets physiques, emprunte *matériellement* le chemin des mathématiques.

Simuler avec l'ordinateur, c'est alors utiliser l'algorithme à la place de la formule mathématique, en tant que moyen de prédiction pour lui faire prédire les uns après les autres tous les éléments de l'avenir que le modèle renferme potentiellement. La simulation en "temps-réel" c'est, chose que le mathématicien n'a pas le temps de faire, prédire tous les instants consécutifs des phénomènes conséquences à un rythme égal à celui auquel ils surviendraient en réalité, et en tenant compte, à ce rythme également, de l'évolution des phénomènes qui les provoquent.

C'est ainsi que l'ordinateur, doté de ses opérateurs numériques, de ses fonctions et de ses transducteurs, autorise ce chemin inverse, du *modèle au phénomène*.

[1] C'est ce concept qui manquait d'ailleurs aux "automaticiens" du XVIIIe siècle. En effet, les tambours à picots qui soulevaient les bras, mouvaient les membres et faisaient jouer les articulations des androïdes contenaient l'information relative au mouvement à effectuer en même temps qu'ils communiquaient, depuis une source centrale, l'énergie nécessaire à ces mouvements.

Enfin, dans une simulation à l'aide de l'ordinateur on rencontrera bien évidemment toutes les conditions évoquées plus haut. L'ordinateur ne leur échappe pas et si l'on considère le niveau le plus fin, celui des opérateurs binaires élémentaires, tests, copies de registres, de mémoire etc. nous parlons de calcul, de logique, de fonctions, mais ces opérations sont des "processus phénoménologiques" : une table de vérité, ou plutôt un circuit "ET", "OU" etc. réalisent une correspondance prédéfinie entre la valeur des digits à ses entrées et celle de sa (ses) sorties.

La "Simulation par Modèle Physique"

Quant à la simulation numérique par modèles physiques, on peut en proposer maintenant une définition circonstanciée : il s'agit d'une simulation dans laquelle l'analyse structurelle d'un objet physique sert à créer un modèle mathématique qui devient la base de la construction d'une structure de processus numériques convergeant vers la simulation globale de cet objet.

La simulation des objets du monde physique à l'aide de l'ordinateur, possible avec une certaine pertinence à partir d'un stade suffisant de ses performances, inaugure alors une forme de représentation sans équivalent antérieur. Les objets représentés ne sont plus seulement observables, mais manipulables, et, plus loin, ce qui est en représentation n'est pas seulement la surface, l'apparence des choses, mais une part de leur être interne, comme si on allait pouvoir le redécouvrir. L'ordinateur permet, dans la quête de la représentation, un déplacement fondamental d'une science et d'une connaissance des phénomènes sensibles vers une science et une connaissance de leurs causes génératrices que l'on met elles-mêmes en représentation.

Toutefois, une simulation par modèle physique n'est jamais entièrement physique et comporte nécessairement des constituants fonctionnels et phénoménologiques, soit en raison des limites de l'analyse (modélisation) soit en raison de celles de la synthèse. Plus loin, la simulation par modèle physique à l'aide de l'ordinateur met en jeu un système de construction dont les plus petits constituants sont des micro-systèmes finis : le nombre d'états dans lesquels ils peuvent se trouver est déterminé par le nombre et la taille des mots binaires qui en permettent la mémorisation et la transmission. Par ailleurs, le dispositif final n'est, lui aussi qu'un agencement fini de ces constituants et le nombre d'états dans lesquels il peut alors se trouver à son tour, aussi gigantesque soit-il, est également fini. L'ordinateur, à ce titre, non seulement ne nous fait pas échapper à la finitude, à la coupure ontologique, mais la consacre d'une façon jusque là inégalée.

Il ne faudrait pas voir du pessimisme dans cette dernière remarque, ni l'amorce d'une conclusion négative. Au contraire. Mais pour comprendre où se trouve le positif, il faut encore quelques remarques. Nous avons tenté de dire ce qu'était la simulation et comment elle se pouvait. Il nous reste, pour finir à essayer de comprendre pourquoi on simule.

TROISIEME PARTIE⊡SIMULER POUR CONNAITRE

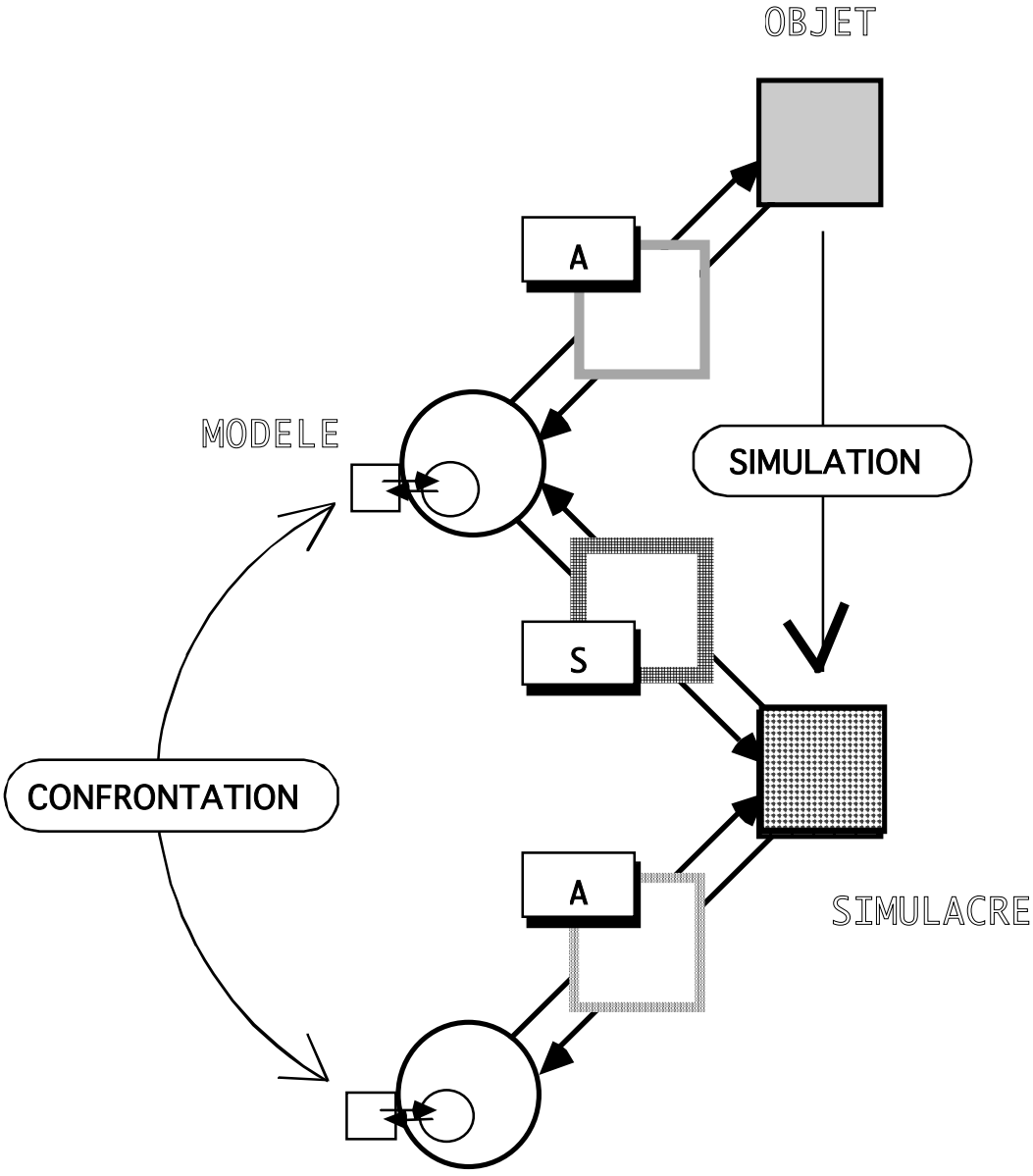
Nous en venons plus significativement maintenant au lieu de rencontre annoncé au début de notre deuxième partie : deux catégories au moins de sujets humains ont en effet recours aujourd'hui à la simulation : les scientifiques de toutes disciplines, et les artistes, en particulier, pour ces derniers, ceux qui créent des images ou des sons par le moyen de l'ordinateur. Considérons, afin d'établir plus aisément des rapprochements, dans les communautés des deux catégories précédentes, les représentants particuliers et en liaison naturelle suivants : les physiciens de l'acoustique et les musiciens. Ce qui est d'emblée frappant, c'est que malgré la différence manifeste de leurs objectifs respectifs, ils sont les uns et les autres partie prenante de la simulation, entre autre de la simulation par modèle physique. La question que l'on peut se poser est alors : qu'est-ce qui les rapproche, qu'ont-ils en commun, et qu'est-ce qui les différencie ?

Observons alors tout d'abord séparément les méthodologies de la simulation chez le physicien et chez l'artiste (informaticien).

1. La simulation du physicien

La simulation, avons-nous dit, correspond à un retour à l'expérimentation, mais à une expérimentation non pas en vraie grandeur ni avec les vrais phénomènes, mais avec un objet ou dans une situation de substitution. Une première différence entre le physicien et l'artiste est alors que si le second a un besoin impérieux du phénomène sous sa forme sensible, le physicien semble pouvoir se contenter d'un "relevé phénoménologique", d'une médiation sous forme de nombres ou de graphiques.

Mais si le physicien simule un objet dont il a déjà par ailleurs une certaine approche, une certaine connaissance, c'est qu'il envisage par là même, d'acquérir une nouvelle maîtrise. Ce faisant, il met alors en oeuvre une méthodologie générale assez complexe, un protocole que l'on peut représenter dans un diagramme en plusieurs étapes :



Tout d'abord, par des moyens matériels expérimentaux et des moyens conceptuels divers, le physicien élabore un modèle, un modèle mathématique de l'objet. Quelle que soit la situation, l'objectif ou le contexte, cette première étape est incontournable. Partant de cette première connaissance, il construit ensuite un algorithme qu'il traduit en programmes. Associés aux transducteurs adéquats, ces processus constituent alors le simulacre.

A l'issue de l'opération, on se trouve ainsi devant deux entités, l'objet et le simulacre, toutes deux a priori abordables selon la même approche expérimentale. Dans la démarche du physicien, une étape est alors fondamentale : celle de la confrontation entre l'original et le simulacre. Cette confrontation, le physicien la mène loin de l'"imperfection" et des caprices des sens, au niveau seul rigoureux et admissible pour lui, du relevé phénoménologique, de la mesure, et par là, il établit la validité du simulacre, c'est-à-dire l'aptitude de ce dernier à se substituer légitimement à l'original, dans les expérimentations. Par essence, puisque l'on n'a de l'original qu'une connaissance définie, qu'il nous est inaccessible au delà d'un certain domaine, la confrontation ne peut évidemment porter que sur ce domaine, mais il peut néanmoins s'engager un certain processus dialectique comportant un retour à la construction (synthèse) voire à la modélisation même (analyse), jusqu'à obtenir une coïncidence satisfaisante entre ce qui est expérimentable simultanément sur l'un et sur l'autre.

Dans cette première phase de la simulation, on s'est en fin de compte fixé un premier objectif : faire coïncider au mieux les aspects phénoménologiques "connaissables". L'identité de ce qui peut être confronté donne alors au physicien confiance en son simulacre, par la même en toute la chaîne analyse/synthèse et, dans la mesure où il est peu probable qu'une dérive en analyse soit exactement compensée par une dérive en sens inverse en synthèse, une certaine confiance respectivement et séparément en ses moyens d'analyse, ses modèles et ses moyens de synthèse. A ce titre, le tout premier produit de la simulation, quand elle est réussie, est une maîtrise, une constitution des outils de cette connaissance. Il n'y a là rien de véritablement nouveau si l'on considère que l'on peut appliquer le même raisonnement au niveau du simple dipôle objet/modèle, sans aller chercher la simulation : si le modèle permet de prévoir effectivement, c'est qu'il possède une certaine vertu de représentation en laquelle on peut avoir confiance…

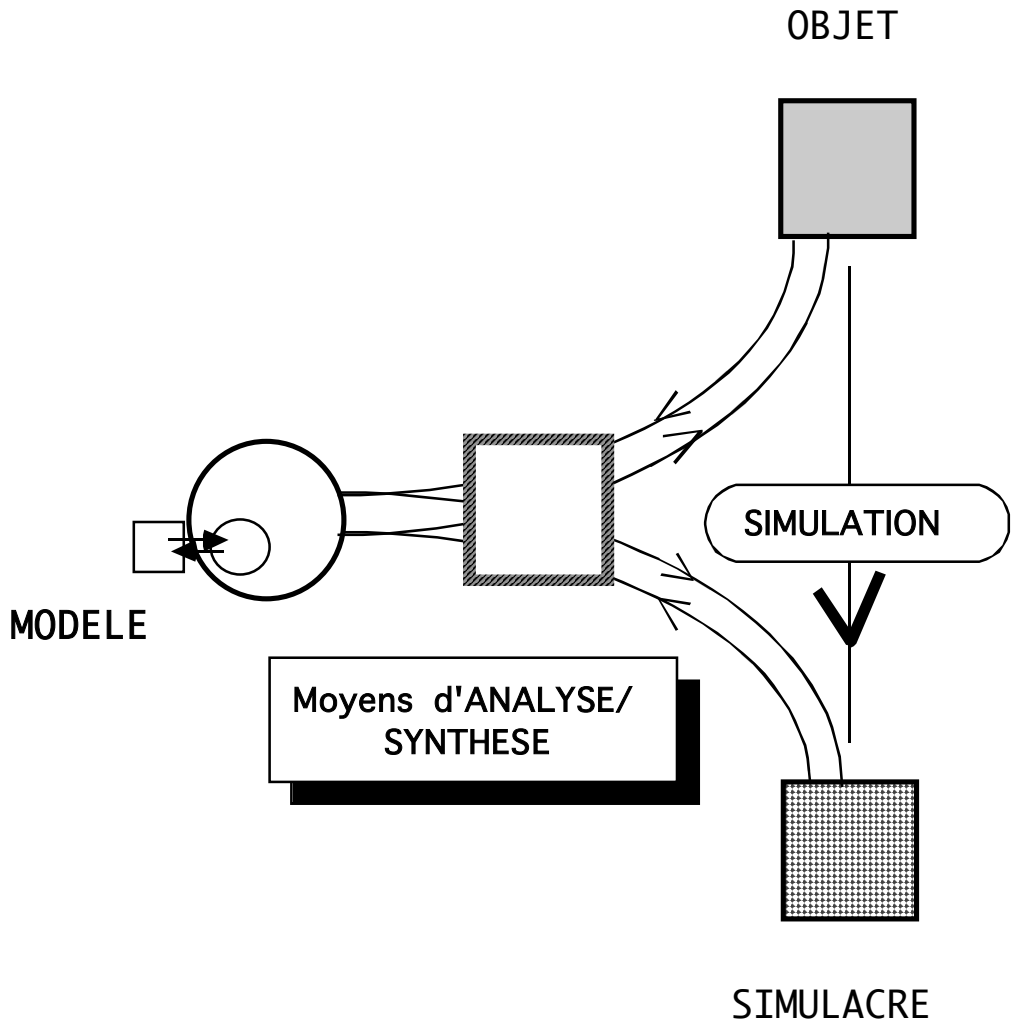
Quel est alors le véritable but, le véritable apport de la simulation pour le physicien ? Dans la mesure où le simulacre, en particulier lorsqu'il est construit dans l'espace discret et fini de l'ordinateur nécessite une connaissance entièrement prédéterminée et également finie de l'objet, quelle nouvelle connaissance peut-on espérer acquérir ? Quelle est la vertu du cercle qui semble se fermer ici ?

2. Une Physique Algorithmique

Il y a une différence essentielle entre le modèle et le simulacre. Le simulacre opère, il est un algorithme au sens fort de ce terme, c'est-à-dire de séquence de processus opérants, descriptibles dans l'abstrait avant d'être construits et effectifs, mais effectifs indépendamment, on pourrait dire plus fortement "malgré", "en dépit" de toute description logique, formelle, mathématique. De cette différence, et dans la mesure où l'algorithme et la description formelle ne réduisent pas l'un à l'autre, il y a, en substance, dans la simulation, plus exactement dans l'algorithme de simulation, une nouvelle forme de représentation et de connaissance, une nouvelle forme d'*explication*, sachant qu'aucune explication absolue et définitive n'est envisageable (tant que notre connaissance des choses n'est qu'une connaissance partielle d'une partie des choses, il ne peut y avoir, pour chaque chose que des connaissances multiples).

Partant de là, incidemment, une nouvelle perspective épistémologique s'amorce. On a évoqué une correspondance, une transposition entre les moyens de l'analyse et ceux de la synthèse.

On a essayé de "faire coller" les seconds aux premiers, et l'on y était contraint parce qu'implicitement on ne pense, au moment de la modélisation, qu'aux outils mathématiques traditionnels, issus par exemple du calcul infinitésimal, dont on cherche une "traduction" dans les nouveaux moyens. Or, si l'on parvient, dans le domaine de représentation propre à l'ordinateur, c'est-à-dire essentiellement celui du discontinu, des tables de vérité, de la mémoire discrète phénoménologique, etc. à créer un système (au sens générateur du terme, c'est-à-dire capable de représenter un très grand nombre de situations à partir d'une base finie et cohérente d'éléments), alors les éléments et les règles de ce système peuvent devenir les moyens mêmes d'une nouvelle forme d'analyse et de description, par essence parfaitement symétrique de la synthèse. Ces moyens d'analyse peuvent alors fonder une véritable physique : une *physique algorithmique* .



C'est dans cette perspective que la démarche CORDIS-ANIMA[1], qui vise à constituer un système de *modélisation* ET de *simulation* des objets physiques tente de s'inscrire.

Enfin, il y a une vertu de prédictibilité propre à la simulation, ou à la physique algorithmique si celle-ci peut s'instaurer, dont il convient de montrer l'irréductibilité à tout modèle mathématique et qui est si fondamentale qu'elle qu'elle est à même, comme le souligne Atlan[2] de modifier les termes de l'éternelle querelle du déterministe récemment ravivée par les "nouvelles alliances", "théories du chaos" et autres "théories des catastrophes"[3].

Citons la réflexion d'Atlan :
" *Parmi les expériences que nous pouvons faire aujourd'hui du déterminisme et de l'aléatoire, il en est une particulièrement intéressante en ce qu'elle est relativement récente et qu'elle illustre au plus près la réalité de cette contradiction (...). Il s'agit des expériences de simulation par ordinateur où deux aspects sont particulièrement frappants. L'un est la fabrication de suites aléatoires à l'aide de programmes utilisant des algorithmes parfaitement déterministes. L'ordinateur génère une suite de chiffres telle que divers tests ne peuvent trouver aucune corrélation entre eux et que la suite obéit ainsi à tous les critères permettant de la considérer comme aléatoire et imprédictible. Or, il se trouve que l'ordinateur fonctionne suivant un programme déterministe tel que, connaissant les conditions initiales, la suite peut être parfaitement déterminée et prédite (...mais par aucun autre moyen qu'en faisant exécuter ce programme par l'ordinateur - il n'existe pas d'algorithme plus court pour réaliser cette suite) (...) Le caractère aléatoire ou déterministe dépend bien évidemment du lieu où se place l'observateur pour en juger : déterministe s'il observe l'algorithme au niveau de son fonctionnement ; aléatoire s'il observe la série produite et y applique des tests d'autocorrélation et de prédictibilité susceptibles d'y découvrir un ordre.* "[4].

Dans cette formulation, la *série de nombres* figure pour ce nous appelons ici le phénomène, un phénomène dans une forme spécifique de représentation. Les *tests d'autocorrélation* et plus généralement de *prédictibilité* figurent pour tout ce qui peut être tentative de prédire par un autre moyen que l'exécution effective de l'algorithme. Ainsi, le physicien peut-il avoir un modèle parfaitement déterministe de son objet, traduit sans aucune perte dans un algorithme, si, de cette manière, il tient alors toute l'information, toute la substance de son objet entre ses mains, peut-on dire, il n'aura aucune autre façon exacte et complète d'y accéder que celle de faire exécuter l'algorithme, de faire se manifester le phénomène.

Et enfin, le physicien, bien qu'il s'en défende parfois farouchement est nécessairement, inévitablement, mais nous y reviendrons plus loin quand nous aurons donné un certain sens à ce terme … artiste.

---

[1] CORDIS-ANIMA est le nom d'un langage et d'un système de modélisation et de simulation développé depuis le début des années 80 à l'ACROE, à Grenoble, par Claude CADOZ, Annie LUCIANI et Jean-Loup FLORENS. Voir aussi l'article portant ce nom dans cet ouvrage.

[2] Henry ATLAN : "Postulats métaphysiques et méthodes de recherche", in "La querelle du déterminisme", collectif, Paris, Gallimard, 1990.

[3] op.cit., et aussi :
Illia PRIGOGINE et Isabelle STENGER, "La Nouvelle Alliance", Paris, Gallimard, 1979
René THOM, "Modèle mathématique de la morphogénèse", Paris, Bourgois, 1980
Ivar EKELAND, "Le Calcul, l'Imprévu. Les figures du temps de Kepler à Thom, Paris, Ed. du Seuil, 1984

[4] op.cit.

3. La simulation de l'artiste

Tout d'abord, pensera-t-on, les artistes ne sont des "simulateurs" que dans ce cadre très spécifique et encore suspect de l'informatique, de l'informatique musicale, de l'image de synthèse, l'image animée par ordinateur etc. Et, même dans ce cadre, tout le monde ne simule pas et quand la simulation intervient, ce n'est qu'au niveau de la création de matériaux, formes et phénomènes élémentaires qui entrent dans la composition de structures plus complexes élaborées selon d'autres paradigmes.

Certes ! Et pourtant, quand, au moment d'introduire la simulation, on a tenté de formuler ce qui la caractérise, on a dit qu'elle était une forme de représentation résolument attachée au niveau phénoménologique. Peut-on donner une caractérisation différente de la création artistique en général ? Y-a-t-il une façon plus précise d'introduire ce qu'il y a de commun à toutes les démarches de création artistique que de dire qu'elles visent, par tous les moyens possibles, à créer des objets dont la seule valeur finale se présente à nous par l'expérience sensible ?

Toute affirmation a ses butées, mais il y a toujours dans ses résonances quelque effet heuristique. C'est alors le moment ici de jouer cette petite percussion : *la création artistique en général relève de la simulation* ; et s'il est une chose qui la distingue en premier lieu de la simulation du physicien, c'est qu'en un certain sens, elle est plus exigeante parce qu'elle ne souffre précisément aucune médiation entre le phénomène et sa perception. Pour la création artistique, le sensible, le perçu, sont ultime nécessité ; le son entendu en musique, la couleur, la lumière, la forme vues, le volume pétri en peinture, en sculpture…

Pour la création artistique, le perçu et le vécu sont ultime nécessité parce que c'est par le perçu et le vécu que s'opère la *compréhension artistique*. Plus exactement, le perçu, la perception *sont* cette compréhension, c'est-à-dire qu'ils ne sont pas seulement des intermédiaires, les moyens d'une saisie informationnelle pour aller vers des niveaux internes de la représentation mentale, mais, indissociables de ces derniers, ils sont des constituants de cette représentation. Et si, dans une démarche où, pour connaître et représenter la réduction est une nécessité, son opposé dialectique l'est tout autant. Dans la compréhension artistique, il y a ce nécessaire et complémentaire retour au caractère ontologique infini de l'homme lui-même et en particulier de ses sens.

C'est une des raisons de l'impossibilité pour la science de vivre sans l'art et pour l'art de vivre sans la science.

La nécessité des sens dans la compréhension artistique est du même ordre que l'irréductibilité de la simulation comme ultime moyen de connaître, dans l'exemple emprunté à H. Atlan plus haut. Inversement, le physicien n'acquiert-il pas une réelle compréhension du phénomène qu'il étudie, peut-être la plus ultime qu'il n'obtienne jamais, même s'il s'en défend par une attitude qui relève quasiment du dépit amoureux vis-à-vis de sa discipline, lorsqu'il réussit à s'en fabriquer une simulation…visible, sensible, sensorielle… sensuelle !?

Mais revenons à plus de mesure.

La création artistique est une simulation dans ce premier sens de l'ultime nécessité du perçu. Mais elle est aussi une simulation dans la démarche et dans ses conditions : de la même manière qu'une simulation physicienne, aussi exigeante et rigoureuse soit-elle, recourt à une compromission articulée entre processus fonctionnels, structurels, phénoménologiques etc., dans la création artistique, tous les moyens sont bons (ce qui ne veut pas dire qu'utiliser n'importe quel moyen pour faire n'importe quoi soit bon !) pour arriver à un résultat précis au niveau du phénomène sensible final. La création artistique s'autorise en permanence un jeu entre la règle et son débordement, entre le naturel et le collage, entre la référence et l'artifice. C'est l'exact équivalent de l'irréductibilité de ce que nous avons plus haut appelé le "mur phénoménologique".

On objectera qu'il ne s'agit là que de circonstances propres à la création du matériau, de la microstructure et que la composition, l'organisation en structures complexes, en architectures élaborées n'a rien à réclamer à la simulation.

A la simulation des objets physiques non ! mais s'agissant d'objets d'un autre ordre, la démarche fondamentale reste la même : la règle compositionnelle est un modèle, son application, par le compositeur ou par un algorithme relèvent, comme la simulation, du développement d'un potentiel structurel en une réalisation événementielle. Et comme dans la simulation, la règle bien définie cohabite avec des événements, des articulations, des associations contingentes, c'est-à-dire nécessaires mais dont la logique n'est pas accessible, explicite au moment où elle agit (ce qui n'exclut pas qu'une analyse ultérieure ne la révélerait pas).

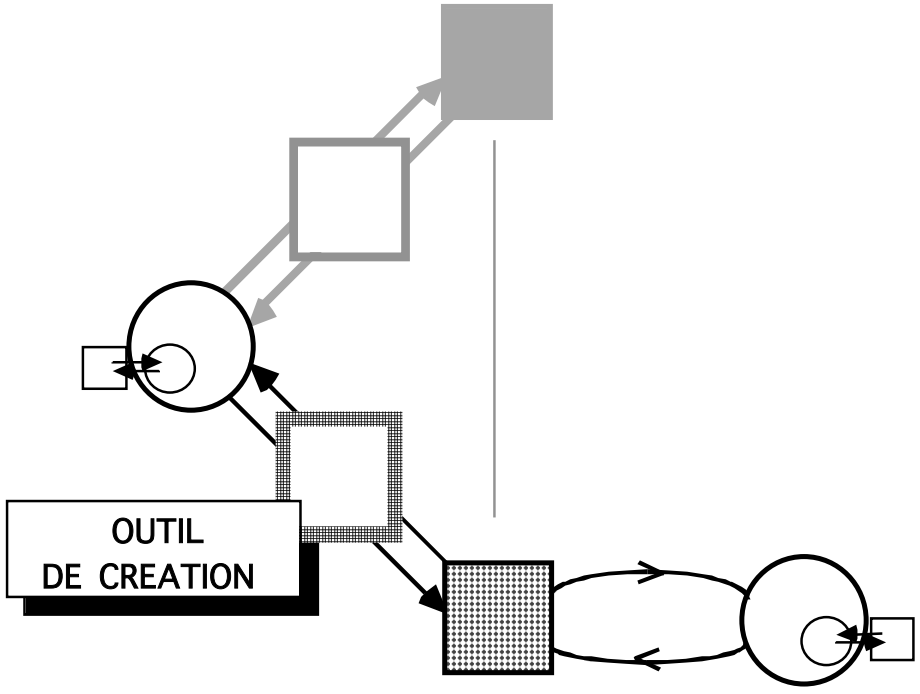
Mais s'il y a une aussi forte proximité entre l'artiste et le scientifique, il faut qu'il y ait aussi une différence.

Il y avait déjà celle de l'attitude vis-à-vis du perçu. Il y en a une plus radicale : la création de l'artiste commence où la démarche du scientifique s'arrête, c'est à dire à l'endroit précis où le simulacre ne correspond plus à rien qui existe préalablement. *L'artiste simule des objets qui n'existent pas* et en cela réside l'acte même de création, le seul qui nous soit autorisé, à nous autres humains, infinis dans notre être, probablement, mais finis dans nos actes.

L'artiste représente des objets qui n'existent pas. Le scientifique, le physicien, puis à leur suite l'ingénieur / entrepreneur usent de la représentation pour connaître la réalité et construire le monde des ressources et des moyens matériels de l'existence. Ils passent eux-aussi par des stades où ce qu'ils représentent n'existent pas encore, mais au moins, c'est pour ensuite concrétiser ces représentations en des objets et des dispositifs réels et utiles.

Si l'artiste simule, c'est bel et bien pour connaître également, et à terme, transformer. Il emprunte simplement un autre chemin : représenter un objet qui n'existe pas et provoquer ce faisant le sentiment essentiel de sa nécessité (que l'on appelle parfois la beauté, mais voila un terme bien pauvre !), c'est convaincre de sa réalité… future, c'est-à-dire convaincre, avant même de connaître le chemin, de s'engager dans sa réalisation. La représentation artistique est plus qu'une prévision, c'est un engagement à faire coïncider le futur avec ce que l'on peut d'ores et déjà souhaiter qu'il soit, c'est faire souhaiter qu'il soit quelque chose.

Avant d'en finir, et pour céder une dernière fois à la logique graphique de nos schémas, on peut figurer, sans plus de commentaires, ce qu'est alors un "outil pour la création artistique" :



Quant à l'ordinateur,

Il radicalise le principe de la simulation et le rend essentiel dans le champ de la représentation scientifique.

Dans le champ de la création artistique il se passe également quelque chose de radical : l'ordinateur nous a permi d'accéder à l'objet d'une quête déjà ancienne : celle de pouvoir représenter les choses non seulement par le verbe, l'image, la forme rigide, ou même la forme articulée, l'image ou le son animés, mais avec leur intérieur, celui qui fait qu'elles apparaissent à nos sens, tout en ayant leurs comportements et leurs façons sous nos actions et nos sollicitations. La simulation par ordinateur pourrait être un nouvel art : un art dans lequel la représentation est plus totale, pas seulement visuelle, audio, et encore moins audio-visuelle, mais… "substantielle".

Avec l'ordinateur, on a ainsi avancé d'un pas fondamental, mais il ne s'agit que d'un pas et nous n'avons pas plus la substance ultime des choses que ne l'avaient les mécaniciens du XVIIIe, et il convient plus que jamais de prendre une mesure exacte : l'ordinateur est un langage au sens où nous l'avons défini ici, c'est-à-dire avec cette condition de finitude. S'il nous fait aller plus loin, il opère également, mieux que tout autre, la "coupure ontologique" qui nous a préoccupé tout au long de ce texte. Prendre les "réalités virtuelles" de nos ordinateurs pour la réalité serait alors instaurer une condition de mort à court terme ; enfermant l'homme dans l'exclusivité de ses rêves, ce serait les rendre et le rendre stériles.

Cela signifie que, comme il en va pour tout langage, celui de l'ordinateur doit impérativement être parlé, c'est-à-dire en permanence replacé en face du réel. Quant à la simulation par modèle physique, il ne faut rien en attendre si on l'envisage, en musique par exemple, comme un substitut définitif aux instruments ou aux véritables objets vibrants. Un simulacre coupé de ses racines ne peut que dépérir, se vider de toute vitalité. Alors que le peintre, aussi abstrait ou symbolique soit-il, retourne toujours au paysage, à la nature, pour dialoguer avec eux par le moyen de son art de représentation, il faut que l'ordinateur, toile où des objets très physiques peuvent venir se déposer en représentation, ne soit à aucun moment coupé de son vis-à-vis, de son horizon. La modélisation et la simulation des objets physiques, des instruments ou des corps vibrants est un dialogue avec le monde réel ; une simulation réalisée est une étape, une attente avant une autre.

S'il y a quelque chose de proprement miraculeux à tenir dans sa main la vibration d'une corde qui n'existe pas, qui n'est que calculs, transducteurs et temps-réel, touchant ainsi, par une nouvelle portée du fictif, à une essence plus profonde du réel, il y a quelque chose de plus miraculeux encore à se rendre compte qu'une corde de violon, un crin d'archet simplement posé dessus, en attente d'un premier frisson, est un univers plus vaste que la plus gigantesque des simulations de l'univers que savent nous proposer les ordinateurs d'aujourd'hui.

Annexe D
Quelques échanges avec les utilisateurs

Notre travail a été structuré en partie par les nombreuses observations d'utilisateurs *in situ* auxquelles nous avons procédé et par les échanges constants et de multiples natures avec certains d'entre eux.

Nous proposons dans cette annexe une copie de quelques échanges de courriers électroniques avec certains des compositeurs qui suivent de longue date le projet : Ludger Bruemmer, Hans-Peter Stubbe et Giuseppe Gavazza.

Ces quelques courriers ne constituent qu'une très petite partie des supports divers ayant servi à l'évaluation du logiciel et plus généralement à la détermination des processus de création que GENESIS avait pour mission de soutenir. Nous les proposons dans la mesure où d'une part ils illustrent en partie le parcours effectué par GENESIS, où d'autre part ils offrent un aperçu (limité, certes) des attentes que les compositeurs peuvent entretenir vis-à-vis de l'environnement et des processus dont il est le support.

1. Giuseppe Gavazza

1.1. Courrier du 01/05/01 – GENESIS 1.3

1.2.

l'objet: Re: Your views on Genesis
De: Giuseppe Gavazza <gigav@iol.it>
à: Peter Torvik <torvik@post.harvard.edu>

Hello Peter,

(…)

Processus de création

1 - My working process on Genesis come on from a few years (first contact wes at a ZKM's Seminar in march '96) and I can define it as "intuitive". I'have a traditional music formation: studies in composition at Music Conservatory and Electronic music and I have a good experience in field of new technologies applied at music. I collaborate for short period in the second half of the 80ths (more with a study approach) at CSC Padua University and LIM (Laboratorio di Informatica musicale) at Milan University; then I worked on compositional projects at Freiburg Experimental Studio in first part ot the 90ths and at Ircam for the cursus annuel 1996-96.
All this experiences were related with an approach not so much "digital" or "science of information-like". As you knowthe early 80ths was the period of the first digital syntesizers and the first personal computer. All this to explain my curriculum b.A. (before ACROE). More in details: with the compositional experience in Freiburg I work with an approach with a strong analog matrix of live electronics. I worked there for some short periods (a few days repeated more time) and this kind of approach was not only the feature af the Freiburg Studio but also my idea about elctronic in music and concert with a more intuitive approach and te possibility to "play" (from mixer consolle) the electronic part during the concert. At Ircam, at the end of the annual cours, january '97 I compose a piece for Soprano, MIDI piano, Diklavier and live electronics. Here it was always the live electronic choice (with no syntesized sounds) but with a strong part of software programmation.

Choice of live electronics was relied to a personal "unlove" for syntetic sounds expecially in concert situations: all kind of way to produce sounds with analog or digital syntesis produce sounds, seems to me, not musically interesting. Later I understand it was also a problem of approach to the way of producing the sounds, the interface between man and software, the way to input the essential data. Work on parameters of sound synthesis is not work direct on sound and less work direct on music.
So I falled in love with Genesis and his visual interface and (but it's the same) with the kind of sounds it creates. It was during a seminar in october '95 at Ircam. So I decided to participate to the atelier at ZKM in April '96.

The fist approach to Genesis, the "imprinting", remained constant enough during my work in these years: the approach was and is intuitive,explorative, physical and with a strong part of "hazard". I don't know the "engine" of the program, I know the basic but I've elaborated a experiencial sintaxis of the objets and of the relations between the objects. Only after a long frequentation I understood some simple, basic abstract and theoretical principles of the program

(as example: the relation between elasticity and viscosity as a similar expression of force in relation with distance or with velocity). Finally my approach to Genesis I think it's very similar to approach with reality of a chidren: the theory will come after as explication of experience.

(...)

Espace de représentation

3 - On first approach I had some problem with topological graphic interface in relation with the visual rapresentation of the objects in the window. I try to explain: the idea of mass and elastic link between masses is very clear but it was not immediately clear the" non sensitivity" of elasticity in relation with distance between the masses on the plan of the screen. In other words: Mass2 is connected to Mass1 with Ressort1, it increase distance from Mass1 and the K of Ressort1 augment so that Mass2 change direction of movement and return in direction of Mass1. Why if I put on the plan of the screen Mass1 and Mass2 closed or far the onefromthe other nothing happen in Ressort1? Now the situation is clear and is higly coherent with the basic priciples of Genesis but this 2 dimension rapresentation of reality was not immediate and intuitive.

Projection des modèles dans les phénomènes

A features, that probably for some one can be and is a (big) problem is the difficulty (impossibility) to be sure of the sound results of a structure. Often with a new (or quite different variation of an old) structure (but also sometimes with a weel known structure entrying different parameters), you obtain surprising sounds. Sometimes (often) they are not interesting or ugly, sometimes they are interesting and nice. This kind of feature for me isn't a blamish rather then an interesting feature. Is a way to find sometimes really new sounds and structures.

I stop here; it's not all but is better send you partial information now then more too later.

best regards

Giuseppe

1.3. Courrier du 10/08/02 – GENESIS 1.4

From gigav@iol.it Fri Sep 6 22:10:47 2002
Date: Fri, 30 Aug 2002 15:47:22 +0200
From: Giuseppe Gavazza <gigav@iol.it>
To: Nicolas Castagne <Nicolas.Castagne@imag.fr>
Subject: Re: PhD soutenance / GENESIS

salut Nicolas,

Finalement j'arrive avec quelques consideration: excuse moi pour le retard, j'espere ne pas etre trop en retard. En touts cas congratulation pour le fine de la these.
(…)

GENESIS 1.4

3 - About 1.4: I've worked very less on 1.4 then 1.3. First consideration: the basic palette of G1.3 can be useful, in my opinion, to start with working with Genesis, it can be a good pedagogical approach. This in consideration of great improvements of G1.4. The basis is the same and the philosophy, in my opinion, is fully respected but this "easy" way to work with G1.4 can bring to greater omogeneite in the work (maybe). Par contre, when you know the program and you have your "style" G1.4 is really great.

4 - Some suggestions:
dans Generation : ajouter la possibilite de definir objets par l'outilisateur et de les inserer dans le menu 4.a.2 - pouvoir choisir les elements des strut generee (CEL ou MAS) (moins importan I know, because is simple Substituer apres, mais ....)

Copies

5 - Inserer la possibilite de "coller" de Memo (like PostIt) on the page de visualisation de struct.

Visualisation des paramètres

6 - inserer la possibilite de visualiser les valeurs sans cliquer 3 ° bouton droit

7 - la fenetre "Generation" est dimensionable sur dimensions differentes; ca c'est bien. Les autres pas (encore?)

Paramètres aléatoires

8 - dans Multiplier ou quelque part inserer la possibilite d'introduire des parametre de disomogeneite (random, definible par l'utilisateur, dans des limites +- X %, etc)

Travail de la sélection

9 - inserer la possibilite de selectionner des objets (toujours sur la fenetre noir) par des figures differentes du rectangle (ronde-ellipse); maybe pas tres interessante.

Pour today is enough,

A la prochaine fois et pardon pour the mixed language

ciao

Giuseppe

2. Hans Peter Stubbe

2.1. Courriers d'Avril 2001 – GENESIS 1.3

```
l'objet: Re: Your views on Genesis
la date: Sun, 22 Apr 01 11:02:27 -0000
De: hans Peter Stubbe TeglbjXrg <stubbe@vip.cybercity.dk>
^: "Peter Torvik" <torvik@post.harvard.edu>

This is just a list a ideas - not a describtion of how I work.
I have also some more high level ideas, but they will have to wait.
I hope this is clear to you what I mean.
Best Regards
Hans Peter
```

Sorties sonores

```
#1) More Outputs: Any power-2 number of tracks is possible in an
aiff-file,
so like wise it should be possible in Genesis.
```

Trans-représentation

```
#2) Textual Export/Import: Make a protocol to format a structure
as simple text. Working both ways, for Export and for Import.
```

Macro-modules / encapsulation

```
#3) Encapsulation: Once having made a f.ex. an excitation-mechanism
that one is satisfied with, the user should be able to encapsulate it
to be used as a new "composite object". The Celbut is an example.
One can store such mechanisms in a seperate file and use "inserer",
but encapsulation - as a form of data-hiding - allows a more
transperant
way of working when building complex structures.
```

Génération

```
#4) The structure generator "generer": Should be made far more
flexible.
Not only the point number of masses, but any parameter m,k,z should
one
be able
to "generate". An idea that demands reflections on type of
appropriate
algorithms.
Using tendency-masks (:choosing randomly between a low and a high
treshhold)
could be one such idea. If low threshold = high then the value will
be
constant.
The text I/O mentioned above can be use for the same purpose.
```

Ouverture du logiciel

```
#5) Function Modules: It would be nice to be able to use user defined
functions in a structure. One very simply, yet much needed, is a
scalor.
One very often encounters the situation that an excitation is too
week.
```

Instead of readjusting its internal parameters, one could simply boost the output (or minimize it if that is what is needed). Other small functions
(that is functions of position or force) would also be very convinient.

Analyse/accordage

#6) More controls on final resulting pitch and duration:
It is possible to tune a structure using "pigmag". Could it be included
in the interface, possibly with a graphical representation of the modal
data? It would be nice if one could tune the damping of a structure,
so that it will sounds only for a user-specified amount of time.
Pitch/Duration controls are paramount important to composers, as you know.

Manipulation directe

#7) Way to much MOUSE: it is a very mouse-heavy environment. Have keyboard shortcuts for as much as possible to lighten the burden.

One other and related idea to light the use of the mouse:
Have an inpector window instead of the cellule info-window:
An inpector window can be open at all time, so the mere pointing,
no clicking and holding, would display the current data.

Conditions initiales

#8) Easier means to specify initial conditions:
Its rather akward to do right now. Maybe include the Position/Speed specification in the module-data window (m,k,z,s + position and vitesse).
It would then do to click a mass to access this information.
One may loose overview, however. Question: how disperted should the data
be?

Best regards

Hans Peter

2.2. Courriers de Mai 2001 – GENESIS 1.3

l'objet: More Ideas on GENESIS ...
la date: Wed, 2 May 01 11:46:25 -0000
De: hans Peter Stubbe TeglbjXrg <stubbe@vip.cybercity.dk>
à: "Peter Torvik" <torvik@post.harvard.edu>

Hello Peter,

I've thought some more about some of the issues we've talked about.

Sorties

1. MORE OUTPUTS/TRACKS
>When you talk about more tracks, are you thinking they should correspond
>to microphones in the current sense of Genesis - that is, would you want each
>track to correspond to the output at a point in the structure, and just be
>allowed to have more than two of them? Or is there something more or else
>that you have in mind? What do you imagine one might do in that case - is

>your interest in spatialization, polyphony, something else?

The last sentence is interessing;
I have been trying to make models of panning in GENESIS (simple spatialisation, lets say). I have used two "seuils" of BUT in mirrors to direct energy two different ways: positive part to the LEFT and negative to the RIGHT. This is not enough though, since I'm cutting out the positive part of the resulting waveform in one channel and vice versa the negative part in the other. Waveforms are always both. The panning could be helpfull in helping certain modes to come out more clearly. In fact SPACE, for me, is not so much a mean of spinning sounds around, rather its a way "to feel the different components of a material", so to speak.

Im my opinion good panning tools allows one for a better investigation/differentiation of a given VS.
And so I would encourage any reflections about this issue.

Trans-représentation

2. TEXT EDITOR
>I had come to the initial conclusion from my work that it would be
>good to have what I have called a "text-based editor" as an
>alternative/secondary way to work on my structures. That is, a non-graphical,
>"expert" view/mode I could switch into where I would see some sort of table or > list of all my points with all their parameters and could simply adjust them directly
>as I wished, perhaps something like an Excel spreadsheet of my points. When you
>talk about a text import and export, I have the idea that you are wanting to do
>the same thing as I am. Do you think so? Or are you wishing to do something else?

Some remarks: It would be fine to have an text-editor based view. From the one could probably easely do the Export/Import I was mentioning.

One could still ask though if Numbers in tables are the most representative of a structure? Just an example: multiple connections from/to a mass how would that be displayed in textform in an understandable way (I mean: it sure is possible, but you'll end up with huge listing, needing a lot of scrolling?)

Actually I was thinking of another graphic-oriented way of editing data: A cellule is a 3-dimensional point (m,k,z), normally asking for a 3D-display. One could include editing into the existing graphical view as following:
Let SIZE represent the value of a mass (click-hold and drag to edit its value)
Let COLOR (or color intensity) represent stiffness (click-hol and drag as above).
Let HORISONTAL DISTANCE between masses represent damping (edit by displacing a mass).
The precise data should be displayed in an inspector window while editing (or when pointing to some element). This way no additional windows are needed.
One could go on and say: let VERTICAL POSITION of a mass control initial position, and something similar for initial speed.
This idea suggest a closer look to the sigificance of the graphical layout.

Génération

3. STRUCTURE GENERATOR
>I don't very well understand what you are saying about tendency-masks,
>maybe you can tell me a little more about that? I understand the idea that
>the structure generator should be more capable, that is clear to me, but the
>rest of that suggestion I did not understand.

I have come to think of another way of defining variable data: Draw an arbibtrary shape (using some mouse-and-key combination), scale it to lie in some range by typing in two numbers. The shape could then be "sampled" to control whatever parameter one wish. Its an very intuitive way, yet still mangeable since you a a range contorl.

Normalisation

4. NORMALISATION:
I think maybe it could be a good idea to have some sort of COMPRESSOR inserted before the normalisation of the 32bit audio files. By compression you can get the low end and high end values into a narrower range, thus when normalising you'll "capture" more of the tiny little things. We all know the situation that one peak is so loud that the remainder is ... silence. This idea should acount for that. Alternatively If not a compressor then some of the many clever bit reduction algorithms that exist.

Contrôle explicite du temps

5. TIME IN GENESIS:
This is just some thoughts for your satisfaction. You have probably noticed that the menu COMPOSER is undeveloped. I think for one thing, its because the notion of "Time" is in some sense absent from GENESIS. And true it is not easy to tell how that should be handled. Maybe have a discussion with Claude about what are the concepts of time in mechanical physics?
Quantifable mechanistic time is represented by the Sampling Rate, but are we aware of its implications on the resulting waveforms? I have often thought of using very much higher SR to capture superfine movements, then changing the SR in the soundfile header afterwards (thus down transposing) to be able to listen. I have never tried it.

I think we need more time tools. For example something that could tell the "arrival time" from a given position/speed plus m/k constellation. Or as I proposed earlier: a mean to scale all dampings so that the would fit into a specified duration.

One also need to think about the hierarchical nature of music: Meters, measures, beats, pulses, etc. I have tried to re-transcribe waveforms from simple double-pendulums into musical notated rhythms with their patterns of acc/rit. I'm not yet done with that, but it seem to me to open up my viev on GENESIS towards a broader compositional world.

I hope these thoughts can be of some inspiration to you.

Best Regards
Hans Peter

3. Ludger Bruemmer

3.1. Courriers de juin 2002 – GENESIS 1.4

```
From l.brummer@qub.ac.uk Fri Sep  6 22:09:48 2002
Date: Thu, 20 Jun 2002 17:27:59 +0100
From: ludger brummer <l.brummer@qub.ac.uk>
Reply-To: ludi <ludi@zkm.de>
To: 'Nicolas Castagne' <Nicolas.Castagne@imag.fr>
Subject: RE: In addition

Dear Nicolas,
I thought quite a while about possible extensions of G1.4. Some
topics come to my mind, some others we have spoken about and some
other ideas are probably not possible inside this system or inside
the paradigms of this system.
So I speak freely as an "innocent person" and you decide what is
worth implementing or what is not possible at all.
```

Filtre en sortie des simulations

```
1. At some point I would need a DV offset remover. It is not that
difficult to implement. I would prefer to have it on output level
after generating the sound. These days I would definetley create
24bit 48 KHz by default. During the file transformation from 32float
to 24bit the DC offset could happen. It should be switchable.
```

Macro-modules, capsules

```
2. I still fancy the idea of a MAX like simplification object: you
simplify an object (a string, or whatever) indicating only the type
of input/output connectors which are defined by the user before. For
example with a special type of "microphone" the user could define the
connections which are than visible on the "simplified" pictures like
a big ball with for example three lines (equivalent to the
microphone) plus a Letter to distinguish between the connection. The
specific advantage would be that one could simplify some parts of the
Patch while others are still accessible. Currently only the a total
zoom can be applied to the patch.
```

Macro-modules, capsules

```
3. As well I think there should be n-channels in the output file
defined by the number of microphones. DC offset remover and
normalisation could be n-looped to process each channel separately.
```

Représentation

```
4. Furthermore I would think a configurable graphic might help
enhancing the understanding of the object. If you want to observe the
behaviour of a specific mass in a complex patch it might help to give
it a bigger shape/different colour, etc.

That's all for now.
If you have some questions about some of the ideas please mail me....
All the best.
Ludger
```

From ludi@zkm.de Fri Sep 6 22:09:57 2002
Date: Wed, 14 Aug 2002 14:46:10 +0200
From: Ludger Bruemmer <ludi@zkm.de>
To: Nicolas Castagne <Nicolas.Castagne@imag.fr>
Cc: "Claude.Cadoz@imag.fr (Claude Cadoz)" <Claude.Cadoz@imag.fr>
Subject: desired options in Genesis ....

Dear Nicolas, Dear Claude
I dont know if you still receive this email. But there was something in my mind after I wrote the last mail regarding Genesis. I think the idea modification of LIAs over time is not so unphysical.

Modification paramétrique - spatialité

Since Genesis has no 3d capabilities there are some effects that cause the physical specification of an object to change. Deformation of objects, either hitting object or vibrating object create a different behaviour of them. For example the hammer of a piano is made out of a rather soft material. If you hit the string very hard the material is compressed and than harder as before.
Maybe you can model this with non-linear Links for a specific state of the material. But the material is aging and or deforming, it gets "tired" or less elastic. So it would get harder after a couple of hits.

I think it should be possible to include the modification of material over time.

all the best

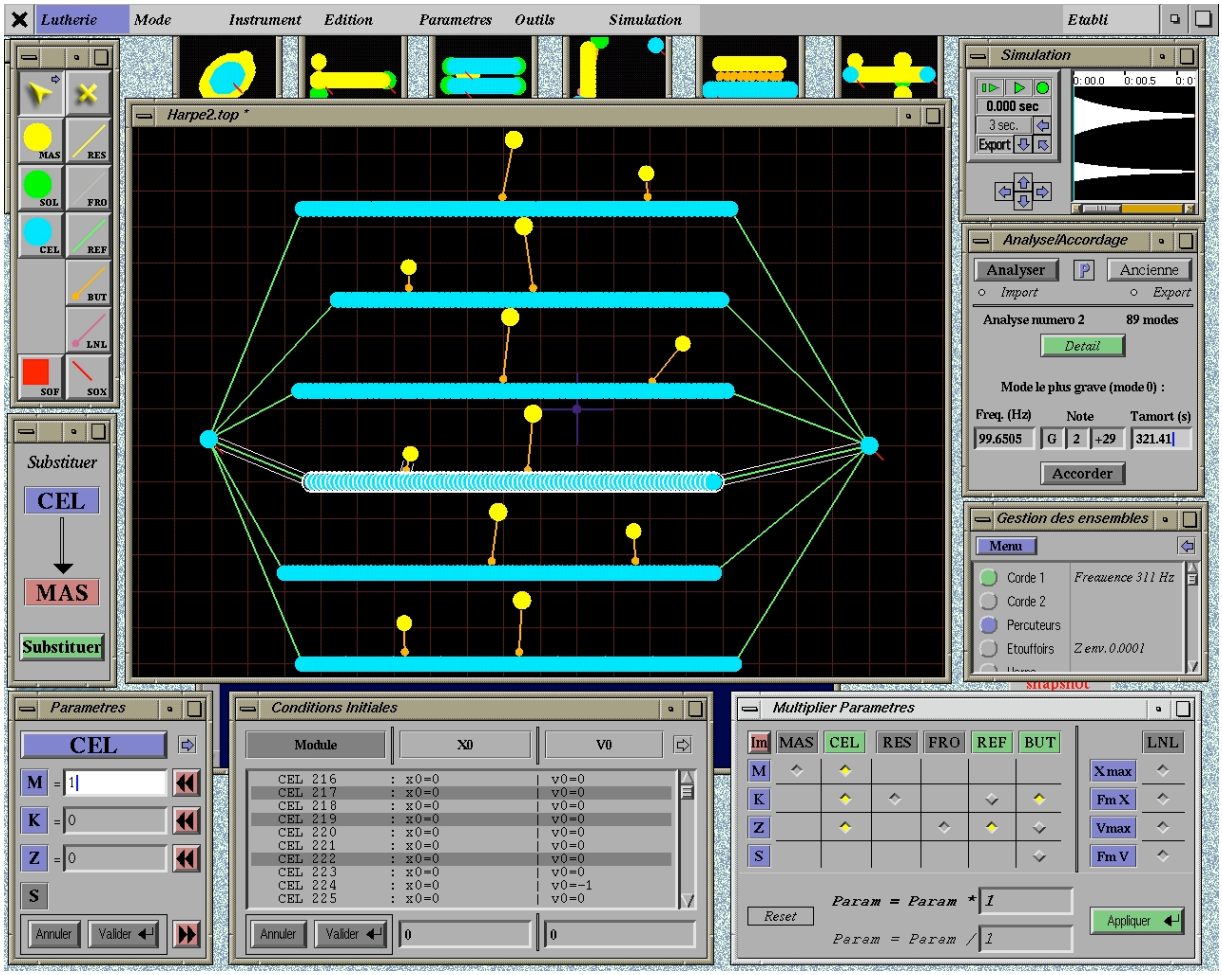
Ludger

Annexe E : Plages couleur

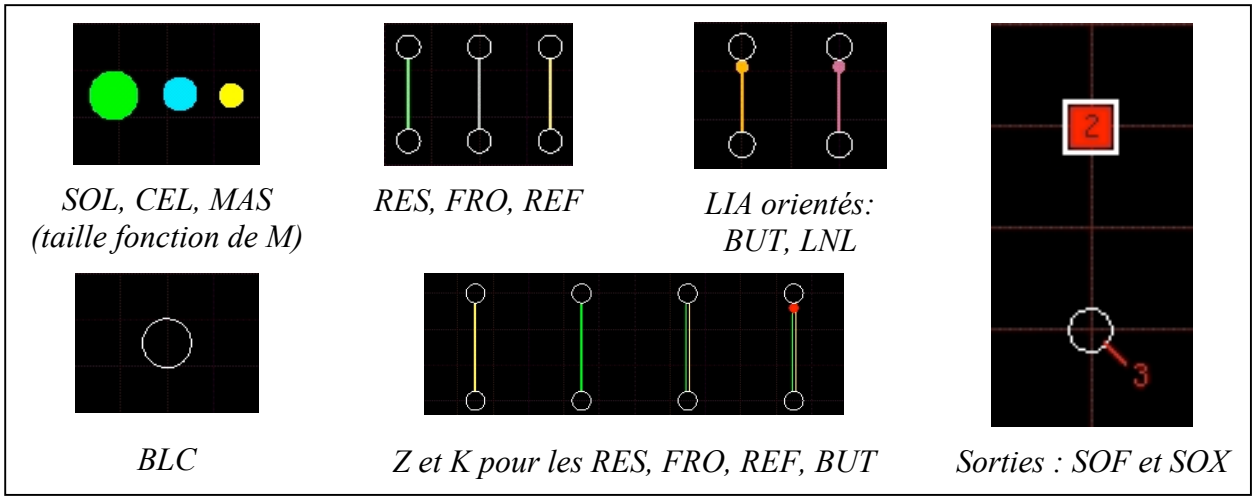
Les pages suivantes reproduisent en couleur les visuels de l'interface présentés dans le corps du texte qui nous ont semblé particulièrement significatifs.

On y trouvera notamment :

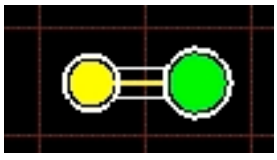
- une vue d'ensemble de l'interface
- des détails quant à la représentation des modules sur l'établi
- des exemples d'objets GENESIS
- un aperçu des principales fenêtre d'interface.



Aperçu général de l'interface GENESIS

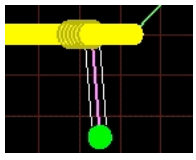


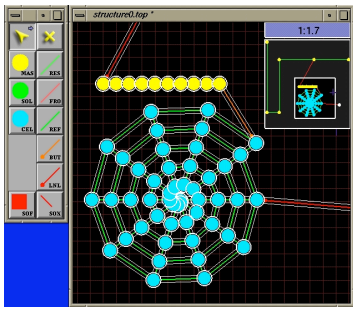
Représentation typologique des modules; représentation des paramètres physiques



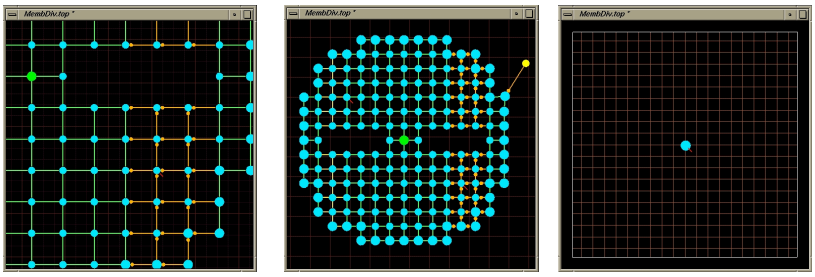
Représentation de la sélection

Désignation

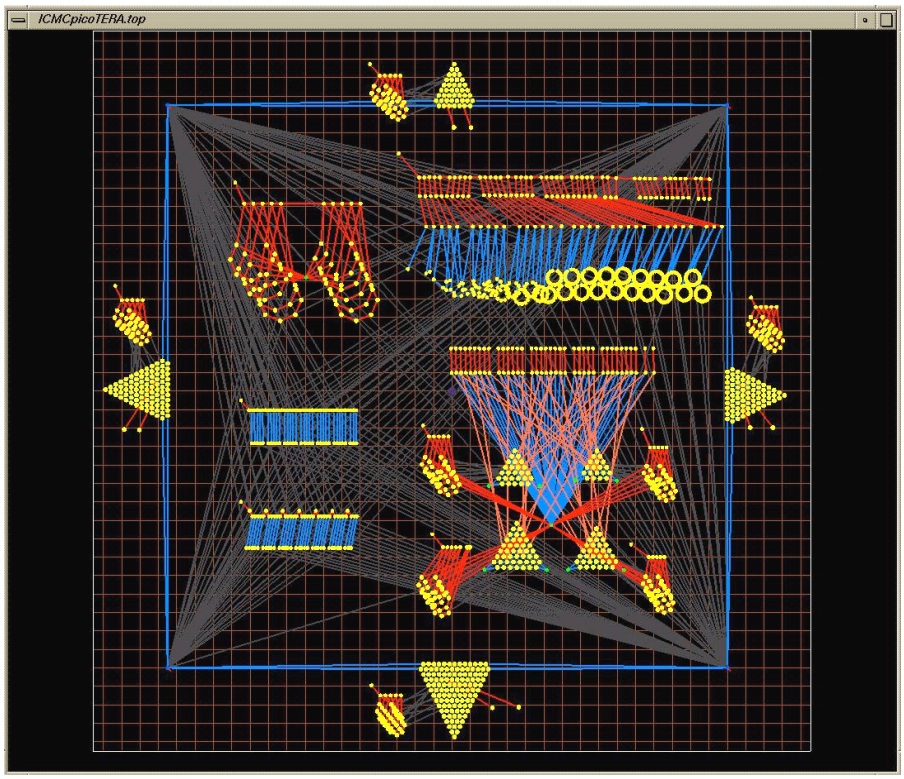




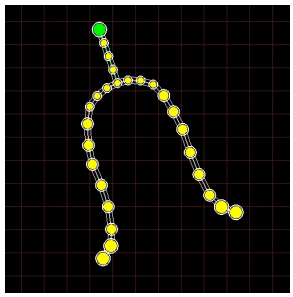
Outil de navigation



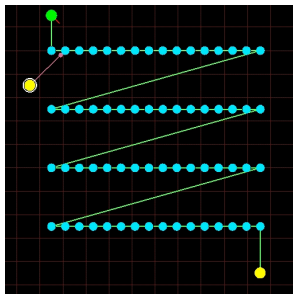
Un modèle à trois taux de zoom différents.
Les modules conservent la même taille.



Disposition libre des modules : la pièce Pico..TERA (C. Cadoz).

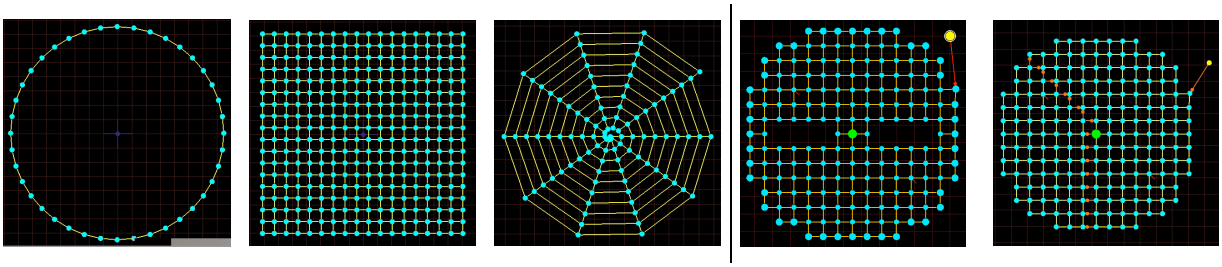


Disposition évoquant l'objet modélisé

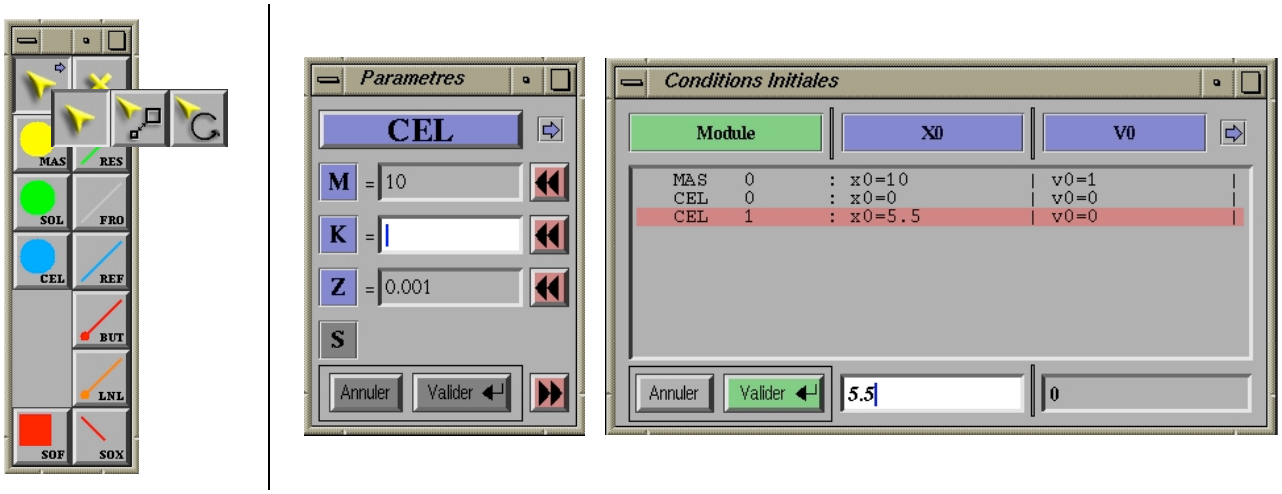


Occupation de l'espace (P. Torvik)

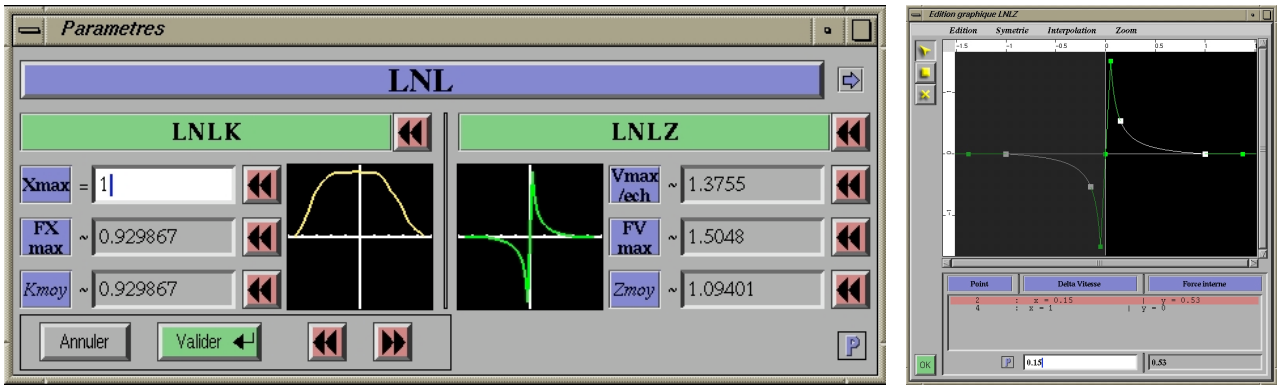
Exemples d'utilisations des deux degrés de liberté offerts sur l'établi.



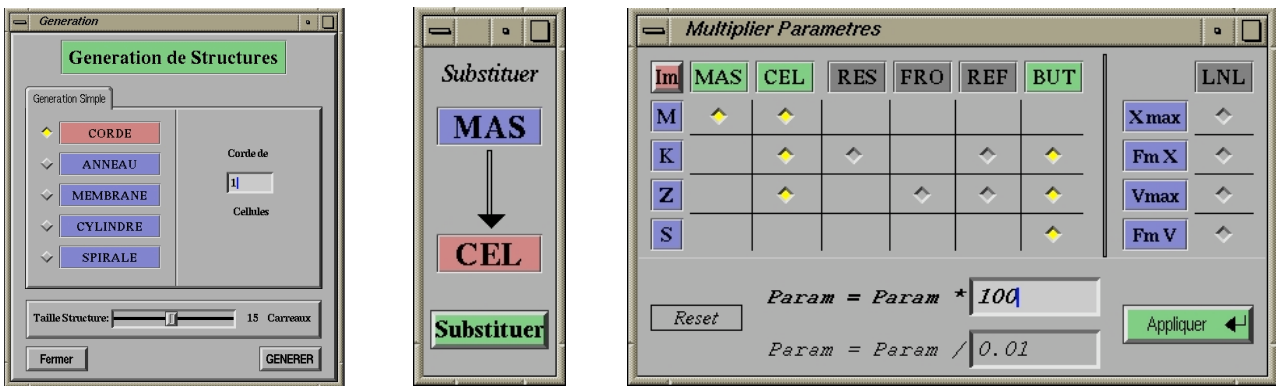
Génération. Gauche : anneau, membranes, spirale. Droit : complexification progressive des propriétés physiques d'un objet généré : substitution de modules.



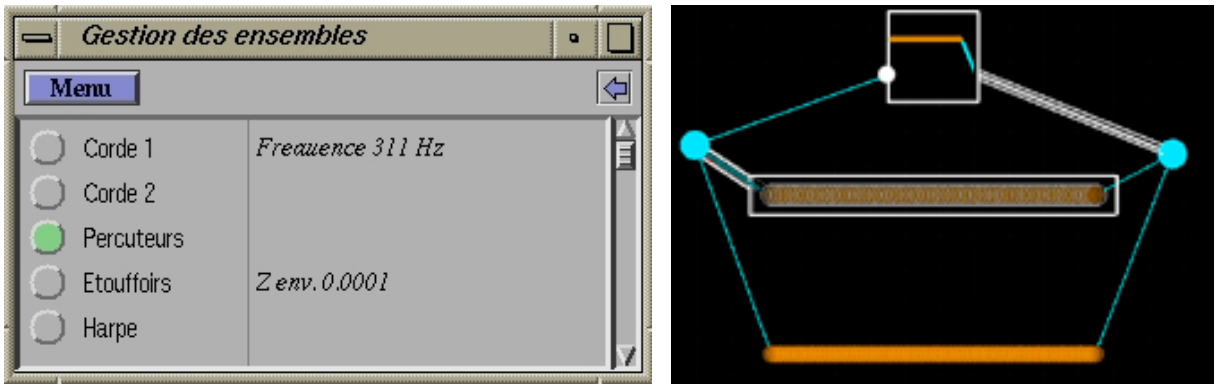
Palette d'outils (gauche).
Fenêtre d'édition des paramètres (centre) et des conditions initiales (droit).



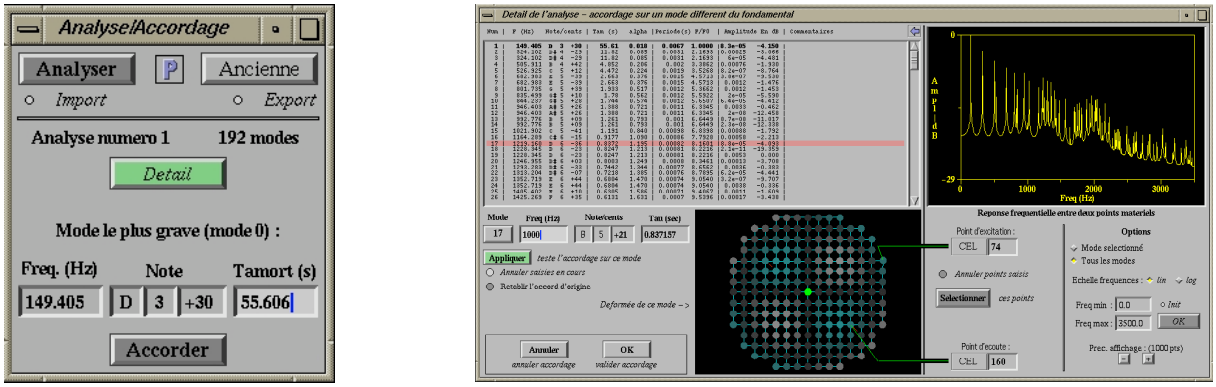
Edition des liaisons non-linéaires (LNL). Amplitude (gauche), forme (droit).



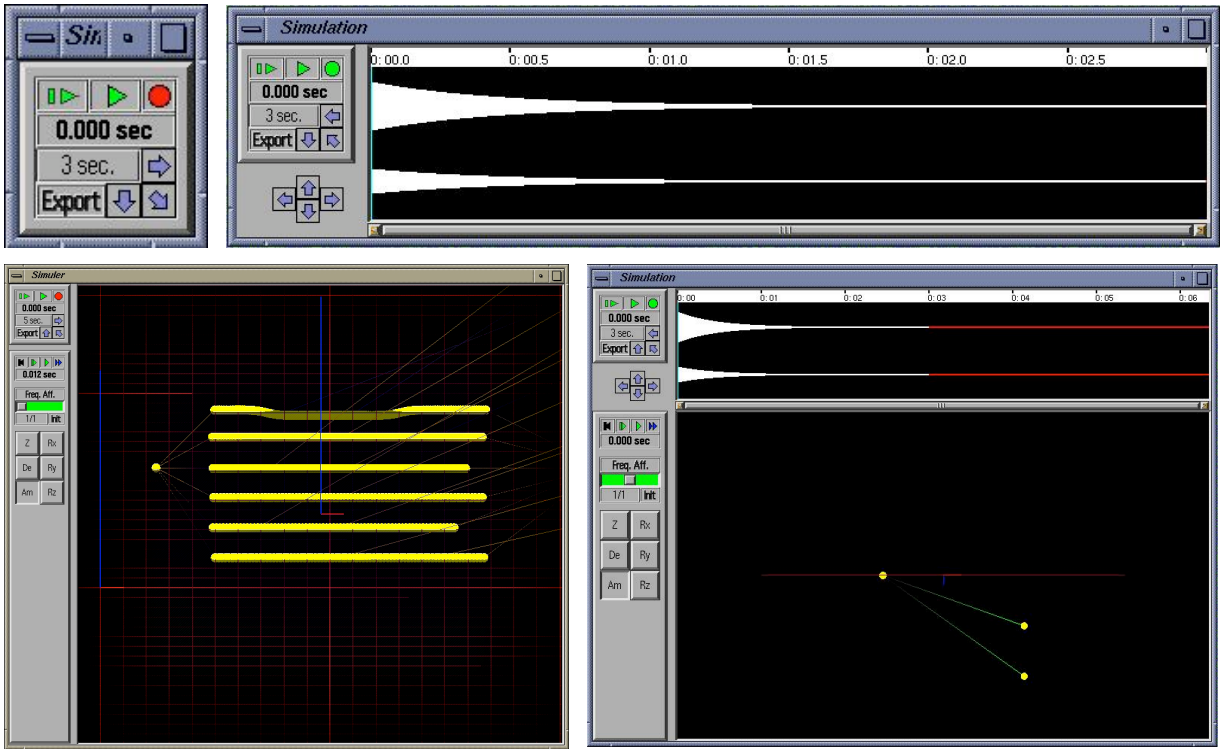
Outil de génération (gauche), de substitution (centre),
de multiplication des paramètres (droit).



Gauche : édition des ensembles de sélection. Droit : représentation des capsules.



Outil pour l'analyse modale. Simple (gauche). Détail (droit, fenêtre modale)



Différents aspects de la fenêtre de simulation

Références bibliographiques

REFERENCES CITEES

Note : les références citées dans les annexes sont incluses dans la bibliographie.
Nombre de références : 145.

Abréviations :

ICMC	: International Computer Music Conference
CMJ	: Computer Music Journal
JAES	: Journal of the Audio Society of America
DAFx	: COST-G6 Conference on Digital Audio Effects
JIM	: Journées d'Informatique Musicale

[Adrien.89] **Adrien JM** : *Simulation numérique de systèmes vibrants complexes. Application à la synthèse sonore par modélisation physique* – Thèse de Doctorat en Acoustique Physique – Université Paris VI - Juin 1989.

[Adrien.91] **Adrien JM** : *The missing link: Modal synthesis* – in G. de Poli, A. Picalli & Curtis Roads: *Representations of musical signals*, MIT Press 1991, S. 269-297

[Agon.98] **Agon Amado CA** : OpenMusic : *un langage visuel pour la composition assistée par ordinateur* – Thèse de Doctorat de l'Université Paris VI – 1998.

[Assayag&Cholleton.95] **Assayag G, Cholleton JP** : *Musique, nombres et ordinateurs* – La Recherche n°26 - juillet-août 1995.

[Baecker&al.95] *Readings in Human Computer Interaction : Toward the Year 2000* – Baecker R., Grudin J., Buxton W.A.S., and Greenberg, S., ed. – Moran Kaufmann, San Fransisco, 1995.

[Baecker&al.95a] **Baecker R Grudin J, Buxton WAS., Greenberg S** *: a Historical and in Intellectual perspective* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.1 pp. 35-53 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed. – Moran Kaufmann, San Fransisco, 1995.

[Baecker&al.95b] **Baecker R Grudin J, Buxton WAS., Greenberg S** : *The Emergence of Graphical User Interfaces* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.1 pp. 49-52- Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Baecker&al.95c] **Baecker R Grudin J, Buxton WAS., Greenberg S** : *Design and Evaluation* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.2 pp.73-91 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Baecker&al.95d] **Baecker R Grudin J, Buxton WAS., Greenberg S** : *Development Tools* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.6 pp. 313-321 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Baecker&al.95e] **Baecker R Grudin J, Buxton WAS., Greenberg S** : *Vision, Graphic Design and Visual Display* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.6 pp. 411-423 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Baecker&al.95f] **Baecker R Grudin J, Buxton WAS., Greenberg S** : *Touch, Gesture and Marking* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.7 pp. 469-481 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Baecker&al.95g] **Baecker R Grudin J, Buxton WAS., Greenberg S** : *Human Information Processing* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.9 pp. 573-585 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Baecker&al.95h] **Baecker R Grudin J, Buxton WAS., Greenberg S** : *From Customable Systems to Intelligent Agents* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.12 pp. 783-792 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Barbaud.66] **Barbaud, P :** *Introduction à la composition musicale automatique*, Dunod, 1966.

[Battier.99] **Battier M** : *L'approche gestuelle dans l'histoire de la lutherie électronique. Etude de cas : le Theremin* – in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[Baudel.96] **Baudel T** : *L'interaction gestuelle : définition, état de l'art et perspectives d'industrialisation* - – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed - Paris, 1996.

[Barnard.95] **Barnard P** : *The Contribution of Applied Cognitive Psychology to the Study of Human-Computer Interaction* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.9 pp. 640-658 - Baeckker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Barrière.90] **Barrière JB** : *Pour une esthétique de la musique avec ordinateur* – Thèse de doctorat de philosophie – Université Paris I Panthéon-Sorbonne - Janvier 1990

[Bayle&al.61] **Bayle F, Chion M, Reibel G** : *les musiques électroacoustiques*, INA/GRM - Edisud, 1961.

[Beaudouin.97] **Beaudouin-Lafon, M** : *Interaction instrumentale : de la manipulation directe à la réalité augmentée* – Actes des Neuvièmes Journées sur l'interaction Homme-Machine, IHM'97 - Cépaduès édition - Poitier, 1997.

[Beaudouin.99] **Beaudouin-Lafon, M** : *Moins d'interface pour plus d'interaction* – in *Interfaces Hommes-Machine et Création Musicale*, pp. 123-141 – sous la direction de Vinet H, Delalande F - Hermes, Paris, 1999.

[Bier&al.93] **Bier E, Stone M, Pier K, Buxton Z et De Rose T** : *Toolglass and Magic Lenses : the see-through interface*. – Acte de la conférence SIGGRAPH, pp. 73-80 - ACM, 1993.

[Bier&al.95] **Bier E, Stone M, Fishkin, K, Buxton W, Baudel, T** : *A Taxonomy of See-through Tools* – in *Readings in Human Computer Interaction : Toward the Year 2000* – Baecker R., Grudin J., Buxton W.A.S., and Greenberg, S., ed. – Moran Kaufmann - San Fransisco, 1995.

[Borin&al.90] **Borin G, De Poli G, Puppin S, Sarti A** : *Generalizing the Physical Model Timbral Class* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Borin&al.92a] **Borin G, De Poli G & Sarti A** : *Algorithms ans Structures for Synthesis Using Physical Models* – CMJ vol 16/4 – MIT Press 1992

[Bosseur.93] **Bosseur D et JY** : *Révolutions Musicales, la musique depuis 1945* Collection Musiques Ouvertes - ed. Minerve - 1993.

[Cadoz.79] **Cadoz C** : *Synthèse sonore par simulation de mécanismes vibratoires, application aux sons musicaux* – Thèse de doctorat de l'Institut National Polytechnique de Grenoble -1979.

[Cadoz.88a] **Cadoz C** : *Informatique et outil de création musicale* – Revue Marsyas n°7 - Sept. 1988.

[Cadoz.88b] **Cadoz C**: *Geste instrumental et composition musicale* – Rapport de recherche ACROE/LIFIA, Grenoble, 1988.

[Cadoz.90a] **Cadoz C** : *Simuler pour connaître, Connaître pour simuler* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Cadoz.90b] **Cadoz C** : *Introduction* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Cadoz.94b] **Cadoz C** : *Le geste, canal de communication homme/machine. La communication instrumentale* – Technique et science de l'information. – Volume 13 - n° 1, pages 31-61 – 1994.

[Cadoz.99a] **Cadoz C** : *Continuum énergétique du geste au son* – – in *Interfaces homme-machine et création musicale* – sous la direction de H Vinet et F Delalande – Hermès, 1999.

[Cadoz.99b] **Cadoz C** : *Musique, gestes, technologie* – in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[Cadoz.02a] **Cadoz C** : *The Physical Model as a Metaphor for Musical Creation* – ICMC'02 proceedings – Sweden 2002.

[Cadoz.02a] **Cadoz C** : *Le Modèle Physique, métaphore pour la création musicale* – Actes des 9e JIM – Marseille, 2002.

[Cadoz&al.81] **Cadoz C, Luciani A, Florens JL** : *Synthèse musicale par simulation des mécanismes instrumentaux, transducteurs gestuels rétroactifs pour l'étude du jeu instrumental* – Revue d'Acoustique, vol. 59, pp. 279-292, 1981

[Cadoz&al.84] **Cadoz C, Luciani A, Florens JL** : *Responsive Input Devices and Sound Synthesis by Simulation of Instrumental Mechanisms : The Cordis System* – CMJ vol 8/2, pp. 60-73 – MIT Press 1984.

[Cadoz&al.90a] **Cadoz C, Luciani A, Florens JL** : *CORDIS ANIMA : Système de modélisation et de simulation d'instruments et d'objets physiques pour la création musicale et l'image animée* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Cadoz&al.90b] **Cadoz C, Lisowski L, Florens JL** : *A Modular Feedback Keyboard Design* – CMJ vol. 14/2, pp. 47-51 – MIT Press1990.

[Cadoz&Ramstein.90] **Cadoz C, Ramstein C** : *Capture, Representation and Composition of the Instrumental Gesture* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – ICMC'90 proceedings – Glasgow 1990.

[Cadoz&Raoult.88] **Cadoz C, Raoult O** : *GENESIS, une interface de contrôle pour la synthèse par modèles physiques* – Rapport interne ACROE – 1988.

[Caelen.96a] **Caelen J** : *Définition et caractérisation des Interfaces Homme-machine* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Caelen.96b] **Caelen J :** *Reconnaître et comprendre la parole* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Card&al.83] **Card S, Moran T, Newell A** : *The Psychology of Human-Computer Interaction* – Lauwrence Erlbaum Associates, 1983.

[Castagne.96] **Castagne N** : *Une exploration des potentialités de la bidimensionnalité dans les modèles CORDIS de structures vibrantes - le filament 2D* – Rapport de stage de DEA Acoustique, Traitement du signal et Informatique Appliqués à la Musique (ATIAM) – et d'Ingénieur de l'Ecole Centrale Paris – ACROE - juillet 96.

[Castagne&Cadoz.98] **Castagne N, Cadoz C** : *Synthèse de sons par modèles physiques : caractérisation et exploration des potentialités de la spatialité des structures vibrantes* – Actes JIM 98 – LMA Publication – Lalonde 1998.

[Castagne&Cadoz.00] **Castagne N, Cadoz C** : *Physical Modeling Synthesis : balance between realism and computing speed* – DAFx-00 proceedings – Italy, December 2000.

[Caussé.90] **Caussé R** : *Non-Linéarités et phénomènes non-linéaires dans les instruments à cordes* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Caussé.93] **Caussé R** : *Morphologie et acoustique du piano* – in Résonance nº 5 - septembre 1993.

[Cavazza.96] **Cavazza M, Constant P** : *Le traitement automatique du langage naturel et ses applications* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Chion.86] **Chion M** : *Une ontologie de la musique concrète* – in *Recherche musicale au GRM* – sous la direction de M Chion et F Delalande – La revue musicale- ed. Richard-Masse - Paris, 1986.

[ColModPhys.90] Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Conversy&Beaudouin.96] **Conversy S, Beaudouin-Laffon, M** : *le son dans les applications interactives* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Corbun&Cadoz.95] **Cadoz C, corbun O** : *GENESIS : Environnement Interactif pour la Création Musicale à l'aide de la Simulation d'Instruments par Modèles Physiques* – Rapport de DEA d'Informatique – ACROE – Ecole National Supérieur d'Informatique et Mathématiques Appliquées – Université Joseph Fourrier – Grenoble 1995.

[Coutaz.87] Coutaz J : *PAC, an implementation model for dialog design* – in Proc. of HCI Interact'87 – pp. 431-436 - Amsterdam, 1987.

[Coutaz.90] **Coutaz J** : *Interfaces Homme-Ordinateur : conception et réalisation* – Dunod, Paris, 1990.

[Coutaz.96] **Coutaz J** : *Ingénierie de l'interaction Homme-Machine* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Coutaz.02] **Coutaz J** : *Architectural Design for User Interfaces* – in *The Encyclopedia of Software Engineering*, seconde édition – J Marciniak Ed. - Wiley & Sons Publ. 2002.

[deLaubier.99] **de Laubier S** : *Le Méta-Instrument a-t-il un son ? Emergence de lois ou de constantes dans le développement d'instruments virtuels* – in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[d'Alessandro&Beautemps.90] **d'Alessandro C, Beautemps D** : Représentation, modification et synthèse du signal vocal par formes d'ondes élémentaires – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Depalle&Rodet.93] Depall P, Rodet X : *de la voix aux instruments* – in Cahiers de l'IRCAM n°2 : *la synthèse sonore* - ed IRCAM 1993.

[DePoli.83] **De Poli G** : *A Tutorial on Digital Sound Synthesis Techniques* – CMJ vol 7/4 – MIT Press1983.

[Djoharian.90] **Djoharian P** : Génération de modèles pour la synthèse modale – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Djoharian.93] **Djoharian P** : Generating Models for Modal Synthesis – CMJ vol 17/1, pp. 57-65 – MIT Press1993.

[Djoharian.97] **Djoharian P** : *L'amortissement dans les phénomènes vibratoires* – Rapport interne ACROE – Grenoble – 1997.

[Dujardin&Vigouroux.96] **Dujardin P, Vigouroux N** : *Les interfaces multimodales : quelles solutions dans les domaines de l'avionique et de l'aide aux handicapés* – in Nouvelles interfaces Homme-machine, publié par l'Observatoire français des Techniques Avancées, série Arago n°18, OFTA ed, Paris, 1996.

[Dufourt.99] **Dufourt H** : *Prolégomènes à la simulation du geste instrumental* – in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[Dufourt&al.98] **Dufourt H, Bovier-Lapierre B, Glas S** – avec la participation de M Decoust et CCadoz : *éléments d'analyse et de documentation - stratégies scientifiques* – in Rapport de mission du ministère de l'Education Nationale - Art-Science-Technologie sous la direction de JC Risset – Tome II – 1998.

[Erickson.95] **Erickson TD**. : *Working with Interface Metaphors* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.9 pp. 147-151- Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995. – Egalement publié dans[Laurel&al.90].

[Florens.90] **Florens JL** : *Modèles et simulations en temps réel de cordes frottées* - 1er Congrès Français d'Acoustique - SFA - Editions de Physique - Lyon Avril 1990.

[Florens&al.98] **Florens JL, Cadoz C, Luciani A** : *A real-time workstation for physical model of multi-sensorial and gesturally controlled instrument* – ICMC'98 proceedings, pp. 518-525 – 1998.

[Fourcade.01] **Fourcade P** : *Etude et simulation des phénomènes percussifs dans les instruments de musique* – Thèse de l'Université de la Méditerranée – Spécialité Acoustique, Traitement du signal et Informatique Appliqués à la Musique – Sept 01.

[Frohlich.97] **Frohlich DM** : *Direct Manipulation and Other Lessons* – in *Handbook of Human-Computer Interaction*, second ed. – pp. 463-487 - Helander M, Landauer TK, Prabhu P, ed. – Elsevier Science BV, 1997.

[Gibet.87] **Gibet S** : *Codage, représentation et traitement du geste instrumental.* – Thèse d'Informatique de l'Institut National Polytechnique de Grenoble, – ACROE-LIFIA-INPG – Décembre 1987.

[Gibson.79] **Gibson JJ:** *The Ecological Approach to Visual Perception* – Houghton Mifflin ed. – Boston, 1979.

[Giraud.99] **Giraud O** : *Architecture logicielle haute performance pour la simulation temps-réel synchrone d'objets physiques multisensoriels : retour d'effort, synthèse de sons, synthèse d'images.* – Thèse de doctorat de l'Institut National Polytechnique de Grenoble – ACROE-CLIPS – Grenoble, juillet 1999.

[Gomoll.90] **Gomoll K** : *Some Techniques for Observing Users* – in *The Art of Computer Interface Design*, p 85. – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Gould.97] : **Gould JD** : *How To design Usable Systems* – in *Handbook of Human-Computer Interaction*, second ed. – pp. 231-245 - Helander M, Landauer TK, Prabhu P, ed. – Elsevier Science BV, 1997.

[Guilbaud.02] **Guilbaud C** : *Modélisation et visualisation de phénomènes naturels simulés par système physique particulaire* – Thèse d'Informatique de l'Institut National Polytechnique de Grenoble – Grenoble, octobre 2002.

[Habibi.97] **Habibi H** : Modèles physiques pour la visualisation d'objets très déformables : Relation, mouvement - forme – image – Thèse d'Informatique de l'Institut National Polytechnique de Grenoble – Grenoble, 1997.

[Heinrich.2001] **Heinrich MN** : *Création musicale et nouvelles technologies, l'instrument représenté ou la relation simulée* – Thèse de doctorat en Sciences de l'Information et de la Communication – Université Stendhal Grenoble III - Avril 2001.

[Hiller&Ruiz.71] **Hiller L, Ruiz P** : Synthesizing musical sounds by solving the wave equation for vibrating objects: – parties I et II - JAES,19-6 et 19-7 – 1971.

[Hirbour.83] **Hirbour L** : *Ecrits et textes d'Edgar Varèse* – Christian Bourgeois - Paris, 1983.

[Hollan&al.97] **Hollan JD, Bederson BB, Helman JI** : *Information Visualisation* – in *Handbook of Human-Computer Interaction*, second ed. – pp. 463-487- Helander M, Landauer TK, Prabhu P, ed. – Elsevier Science BV, 1997.

[Hutchins&al.86] **Hutchins EL, Hollan JD, Norman DA** : *Direct Manipulation Interfaces* – in User Centered System Design, pp. 83-124 – Norman DA, Draper SW, ed. - Lawrence Erlbaum Associates, 1986.

[IBM] Guides pour la conception des systèmes interactifs, http://www.ibm.com/IBM/HCI/guidelines/, ©IBM.

[Incerti.96] **Incerti E** : *Synthèse de sons par Modélisation Physique de Structures Vibrantes. Applications pour la Création Musicale par Ordinateur* – Thèse d'Informatique de l'Institut National Polytechnique de Grenoble – Grenoble, 1996.

[Incerti&Cadoz.95a] **Incerti E, Cadoz C** : *Time-Frequency Decomposition by means of Physical Model* – ISMA proceedings pp. 459-464– Paris, 1995.

[Incerti&Cadoz.96] **Incerti E, Cadoz C** : *Synthesis and Analysis Tools with Physical Modelling : an Environment for Musical Sound Production* – ICMC'96 proceedings - Hong-Kong, August 1996.

[Jaffe.95] **Jaffe DA :** *Ten Criteria for Evaluating Synthesis Techniques* – CMJ vol 19/1 – MIT Press1995.

[Jeffries.97] **Jeffries R** : *The Role of Task Analysis in the Design of Software* – in *Handbook of Human-Computer Interaction*, second ed. – pp. 347-359- Helander M, Landauer TK, Prabhu P, ed. – Elsevier Science BV, 1997.

[Johnson.92] **Johnson P** : *Human Computer Interaction – Psychology, Task Analysis and Software Engineering* – McGraw-Hill Book Compagny - 1992.

[Karplus&Strong.83] **Karplus K, Strong A** : *Digital Synthesis of Plucked String and Drum Timbres* – CMJ vol 7/2 – MIT Press1983.

[Kay.90] **Kay A** : *User Interface: a personal view* – in *The Art of Computer Interface Design*, pp. 191-207 – B. Laurel ed. - Addison-Wesley - New-York 1990.

[Laliberté.99] **Laliberté M** : *Archétypes et paradoxes des nouveaux instruments* – in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[Lascaux.86] **Lascaux T:** *Analyse numérique matricielle appliquée à l'art de l'ingénieur* – Tome 1 et 2 – Masson – Paris, 1986.

[Laurel.90a] **Laurel B** : *Introduction* – in *The Art of Computer Interface Design* – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Laurel.90b] **Laurel B** : *Interface Agents : metaphor with character* – in *The Art of Computer Interface Design,* pp. 347-355 – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Laurel&al.90] *The Art of Human Computer Interface* – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Lindsay&Norman.77] **Lindsay P, Norman D** : *Human Information Processing : An Introduction to Psychology* – Second Edition, Academic Press, 1977.

[Loizillon.96] **Loizillon G** : *Modes de description des sons et synthèse sonore* – Thèse de doctorat de l'Université Paris VII -1995-1996.

[MacKenzie.95] **MacKenzie IS** : *Movement Time Prediction in Human-Computer Interfaces* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.7pp. 483-493 - Baeckker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Manoury.99] **Manoury P** : *Le stade pictographique de la musique électronique doit être dépassé* – in *Interfaces Hommes-Machine et Création Musicale*, – sous la direction de Vinet H et Delalande F - Hermes, Paris, 1999.

[Marcus.95] **Marcus A** : *Principles of Effective Visual Communication for Graphical User Interface Design* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.6 pp. 425-442- Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Marcus.97] **Marcus A** : *Graphical User Interface* – in *Handbook of Human-Computer Interaction*, second ed. – pp. 441-459- Helander M, Landauer TK, Prabhu P, ed. – Elsevier Science BV, 1997.

[McIntyre&Woodhouse.79], McIntyre ME, Woodhouse, J : *On the fundamentals of bowed-string dynamics* - Acustica 43 (2), 93-108

[Morrison&Adrien.93] **Morrison JD, Adrien JM** : *MOSAIC : a Framework for Modal Synthesis* – CMJ vol 17/1 – MIT Press1993.

[Mountford.90] **Mountford SJ** : *Tools and Techniques for Creative Design* – in *The Art of Computer Interface Design*, pp. 17-30. – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Murch.95] **Murch G** : *Color Graphics – Blessing on Ballyhoo* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.6 pp. 442-443 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995.

[Naël.96] **Naël M** : *L'évaluation des interfaces homme-machine interactives* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Neale&Carroll.97] **Neale DC, Carroll JM**. : *The role of Metaphors in User Interface Design* – in *Handbook of Human-Computer Interaction*, second ed. – pp. 441-459- Helander M, Landauer TK, Prabhu P, ed. – Elsevier Science BV, 1997.

[Néel.96] **Néel F** : *L'interaction vocale avec la machine : applications et perspectives* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Nelson.90] **Nelson TH** : *The Right Way to Think About Software Design* – in *The Art of Computer Interface Design*, pp. 235-244. – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Norman.88] **Norman DA** : *The Psychology of Everyday Things* – Basic Book, New-York – 1988. – Réédité en 1990 sous le titre : *The Design of Everyday Things*, Doubleday, 1990.

[Norman.90] **Norman DA** : *Why Interfaces Don't Work* – in *The Art of Computer Interface Design* - p209. – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Olsen.97] : **Olsen DR Jr**. : *Interactive Software Architecture* – in *Handbook of Human-Computer Interaction*, second ed. – pp. 463-487 - Helander M, Landauer TK, Prabhu P, ed. – Elsevier Science BV, 1997.

[Pascal.99] **Pascal M** : *Le studio instrumental : les données d'une virtualité à l'intérieur même du son* – in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[Pearson&Howard.95] **Pearson M, Howard DM** : *A musician approach to physical modeling* ICMC'95 proceedings - 1995.

[Pearson&Howard.96] **Pearson M, Howard DM**: *Recent developments with the TAO physical modeling system* – ICMC'96 proceedings - 1996.

[Polfreman.02] **Polfreman, R :** *MfOM: a Library for Building and Controlling Physical Model Instruments in OpenMusic* – ICMC'02 proceedings – Sweden 2002.

[Razafindrakoto.86] **Razafindrakato, A** : *Le système ANIMA : éditeur d'objets producteurs d'images; implantation d'algorithmes de simulation temps réel* - Thèse d'Informatique de l'université scientifique, technologique et médicale de Grenoble et de l'institut national polytechnique de Grenoble – Grenoble, 1986.

[Rheingold.90] **Rheingold H** : *An Interview with Don Norman* – in *The Art of Computer Interface Design*, p 85. – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Risset.93] **Risset JC** : *Synthèse et matériau musical* – in Cahiers de l'IRCAM n°2 : *la synthèse sonore* - ed IRCAM 1993.

[Risset.90] **Risset JC :** *Modèles physiques et perception, Modèles physiques et composition* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Risset.99a] **Risset JC** : *Evolution des outils de création sonore* – in *Interfaces homme-machine et création musicale* – sous la direction de H Vinet et F Delalande – Hermès, 1999.

[Risset.99b] **Risset JC** : *Nouveaux gestes musicaux, quelques points de repères historiques* – in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[Roads.80] **RoadsC** : *Interview with Max Mathews* – CMJ vol 4/4 – MIT Press1980.

[Roads.93] **Roads C** : *Initiation à la synthèse par modèles physiques* – Cahiers de l'IRCAM n°2 : *la synthèse sonore* - ed IRCAM 1993.

[Rodet&al.90] **Rodet X, Depalle P, Fleury G et Lazarus F** : *Modèles de signaux et modèles physiques d'instruments, études et comparaisons* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Rodet&Vergez.96] **Rodet X, Vergez C** : *Physical Models of Trumpet-like Instruments Detailed Behavior and Model Improvements* – ICMC'96 proceedings - Août 1996.

[Rubine.91] **Rubine D** : *The Automatic Recognition of Gestures* – Thèse de Doctorat - CMU-CS-91-292 – Carnegie Mellon University, 1991.

[Russolo.13] **Russolo L :** *L'art des bruits: manifieste futuriste* – Richard Masse, Paris, 1954

[Sadek.96] **Sadek D** : *Le dialogue homme-machine : de l'ergonomie des interfaces à l'agent intelligent dialoguant* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Salomon.90] **Salomon G** : *New Uses of Color* – in *The Art of Computer Interface Design*, pp. 269-279, – B. Laurel ed. - Addison-Wesley, New-York 1990.

[Schaeffer.66] **Schaeffer P** : *Traité des objets musicaux* – Seuil - Paris 1966.

[Shneiderman.83] **Shneiderman B** : *Direct Manipulation : A Step Beyond Programming Languages* – IEEE Computer, 16(8), p57-69. Aout 1983.

[Shneiderman.98] **Shneiderman B** : *Designing the User Interface. Strategies for effective Human Computer Interaction* – 3rd Edition -Addison Wesley, 1998.

[Smith.96] **Smith III JO** : *Physical Modeling Synthesis Update* – CMJ vol 20/2 – MIT Press1996.

[Szilas&Cadoz.93] **Szilas N, Cadoz C** : Physical Models that learn – ICMC'93 proceesings - Japan 1993.

[TableRonde.90a] Table ronde *Modèle physique pour la synthèse sonore* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[TAO] *TAO Music Synth - UserManual* – Disponible à : http://web.ukonline.co.uk/taosynth - release 2000.

[Torvik.01] **Torvik P :** *Edition évoluée des paramètres dans la synthèse sonore par modèles physiques* – Rapport de stage de DEA Acoustique, Traitement du signal et Informatique Appliqués à la Musique (ATIAM) – ACROE - juillet 01.

[Toulemonde&al.99] **Toulemonde O, Moënne-Loccoz P, Donzel-Gargand B** : *De nouvelles lutheries informatiques* **–** in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[Usabilis] *Utilisabilité web & logiciel* - JF Nogier rédacteur – http://www.usabilis.com .

[Utilisab] *The Usability Methods Toolbox,* brought by James Hom – http://jthom.best.vwh.net/usability/

[Vinet.99a] **Vinet H** : *Concepts d'interfaces graphiques pour la production musicale et sonore* – in *Interfaces Hommes-Machine et Création Musicale*, – sous la direction de Vinet H et Delalande F - Hermes, Paris, 1999.

[Vinet.99b] **Vinet H** : Introduction – in Interfaces Hommes-Machine et Création Musicale, – sous la direction de Vinet H et Delalande F - Hermes, Paris, 1999.

[Wanderley.99] **Wanderley M** : *Contrôle gestuel de la synthèse sonore* – in *Interfaces Hommes-Machine et Création Musicale*, – sous la direction de Vinet H et Delalande F - Hermes, Paris, 1999.

[Xénakis.63] **Xenakis, I :** Musiques formelles – ed. Richard Masse – Paris, 1963.

BIBLIOGRAPHIE ADDITIONNELLE

55 références

[Adorno.62] **Adorno TW** : *Philosophie de la nouvelle musique* – Paris, Gallimard, 1977.

[Adrien&Morrison.90] **Adrien JM, Morrison JD** : *MOSAIC : a modular program for synthesis by modal superposition* – Actes du Colloque *Modèles Physiques, Création Musicale et Ordinateurs* – organisé par l'ACROE, Grenoble 1990 – Edition de la Maison des Sciences de l'Homme – Paris, 1994.

[Assayag.93] **Assayag G** : *CAO, vers la partition potentielle* – Cahiersde l'IRCAM n°3 : *la composition assistée par ordinateur* – Ed. IRCAM, 1993.

[AST.98] *Rapport de mission du ministère de l'Education Nationale « Art-Science-Technologie »* – sous la direction de Jean-Claude Risset – 1998.

[Baggi&al.92] *Readings in Computer Generated Music* – Baggi D editor – IEEE Computer Society Press, 1992.

[Barriere&al.91] *Le timbre, métaphore pour la composition* – sous la direction de JB Barrière – Bourgeois - Paris, 91.

[Borin&al.92b] **Borin G, De Poli G, Sarti A** : *Sound Synthesis by Dynamic Systems Interaction* – in *Readings in Computer-Generated Music* – D Baggi ed. – pp. 139-160 - IEEE Computer Society Press, 1992.

[Bruner.66] **Bruner, J** : *Toward a Theory of Instruction* – – Harward University Press, Cambridge, 1966.

[Cadoz.91] **Cadoz C** : *Timbre et causalité* – in *Le timbre, métaphore pour la composition* – Coordonné par JB Barrière JB – Bourgeois - Paris, 91.

[Cadoz.94a] **Cadoz C** : *Les réalités virtuelles* – Coll. Dominos - ed. Flammarion - 1994.

[Cadoz.96] **Cadoz C** : *Réintroduire les sensations physiques : des interfaces manuelles pour montrer, toucher, palper, éprouver* – n° spécial la Recherche *l'ordinateur au doigt et à l'oeil* - vol. 285 - 1996.

[Cadoz&al.93] **Cadoz C, Luciani A, Florens JL** : *CORDIS-ANIMA : a Modeling and Simulation System for Sound and Image Synthesis: The General Formalism* – CMJ vol 17/1 – MIT Press 1993.

[Castagne&Cadoz.02a] **Castagne N, Cadoz C** : *GENESIS: A Friendly Musician-Oriented Environment for Mass-Interaction Physical Modeling* – ICMC'02 proceedings – Sweden 2002.

[Castagne&Cadoz.02b] **Castagne N, Cadoz C** : *Creating Music by Means of "Physical Thinking": The Musician Oriented Genesis Environment* – DAFx-02 proceedings – Germany, September 02.

[Castagne&Cadoz.02c] **Castagne N, Cadoz C** : *L'environnement GENESIS : créer avec les modèles physiques masse-interaction* – Actes des 9e JIM – Marseille 2002.

[Chemillier&al.97] *Recherches et applications en informatique musicale* – Chemillier M et Pachet F, coordonateurs – Hermès - Paris, 1998.

[Chowning.73] **Chowning J** : *The Synthesis of Complex Audio Spectra by Means of Frequency Modulation* – JAES 21(7) - 1973. – Publié à nouveau dans : – CMJ vol 1/2 – MIT Press 1977.

[Coutaz&Nigay.01] **Coutaz J, Nigay L** : *Architecture logicielle conceptuelle des systèmes itneractifs* – in *Analyse et Conception de l'Interaction Homme-Machine dans les systèmes d'information* - Hermès publ. 2001.

[DePoli&al.91] Representations of Musical Signals – G De Poli, A Piccialli and C Roads ed. - MIT Press - 1991.

[Fatus.94] **Fatus C :** *Vocabulaire des nouvelles technologies musicales* – Coll Musique ouverte - ed Minerve, Paris, 1994.

[Florens&Cadoz.91] **Florens JL, Cadoz C :** *The physical Model : Modeling and Simulating the Instrumental Universe* – in *Representation of Musical Signals* – G De Poli, A Piccialli and C Roads ed. - MIT Press - 1991.

[Fourcade&Cadoz.96] **Fourcade P, Cadoz C** : *sound synthesis by physical modelling: an elementary stricker* – Forum ACOUSTICUM proceedings – 1996.

[Francisco&al.97] **Francisco I, Caussé R et Dudas R** : *Recent work around Modalys and Modal Synthesis* – ICMC'97 proceedings - Septembre 1997.

[Francisco&al.99] **Francisco I, Schnell N et Lartillot O** : *Modalys in jMax: real-time modal-synthesis* – ICMC'99 proceedings - Octobre 1999.

[Genevois&al.99] *Les nouveaux gestes de la musiques* – textes réunis sous la direction de H Genevois et R de Vivo – ed. Parenthèses, coll Eupalinos, Marseille, 1999.

[Greenspan.73] **Greenspan D** : *Discrete Models - Applied Mathematics and Computation - A series of monographs, textbooks, reference works* – Addison-Wesley Publishing Compagny – 1973.

[Griffiths.78] **Griffiths P** : *Histoire concise de la musique moderne, de Debussy à Boulez* – Titre original : *Modern Music* – Coll. les chemins de la musique - ed. Fayard – 1978 – Réédition revue et augmentée : *Brève histoire de la musique moderne* - Coll. les chemins de la musique - ed. Fayard - 1996.

[Guiard.87] **Guiard Y** : *asymmetric Division of Labor in Human Skilled Bimanual Action : The Kinematic Chain as Model* – Journal of Motor Behavior 19(4), pp. 486-517, 1987.

[Hanappe.99] **Hanappe P** : *Design and Implementation of an Integrated Environment for Music Composition and Synthesis* – Thèse de Doctorat de l'Université Paris VI – Spécialité Acoustique, Traitement du signal et Informatique appliqués à la Musique - 1999.

[Haury.99] **Haury J** : *Petite histoire illustrée du clavier* – in *les nouveaux gestes de la musique* – sous la direction de H Genevois et R de Vivo – Ed. Parenthèses, coll Eupalinos - Marseille, 1999.

[Hirbour.83] **Hirbour L** : *Ecrits et textes d'Edgar Varèse* – Christian Bourgeois - Paris, 1983.

[Hofstadter.85] **Hofstadter D** : *Gödel, Escher, Bach : an Eternal Golden Braid* - BasicBooks, 1979.

[Hufschmitt.00] **Hufschmitt A** : *La synthèse par modèles physiques* – mémoire de maîtrise, sous la dir de Marc Battier – Université Paris VI sorbonne - 2000.

[Jaffe&Smith.83] **Jaffe DA, Smith JO :** Extensions of the Karplus-Strong plucked-string algorithm. – CMJ vol 7/2 pp. 56-67 – MIT Press1983.

[Johnson&al.95] **Johnson J, Roberts TL, Verplank W, Smith DC, Irby C, Beard M, Mackey K** : *The Xerox Star: a Retrospective* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.1 pp. 53-70 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann - San Fransisco, 1995.

[Jot.95] **Jot JM, Warusfel O** : *A real-time spatial sound processor for music and virtual reality applications* - ICMC'95 proceedings - Canada, Septembre 1995.

[Lafosse.80] **Lafosse R** : *Ces musiciens et leurs drôles de machines* – R. Lafosse ed. - ArtecSigma, Bordeaux, 1980.

[Luciani.93] **Luciani A** : *Ordinateur, image et mouvement, apparition de l'instrumentalité dans les arts du mouvement visuel* – Cahiers de l'IRCAM n°2 : *la synthèse sonore* - ed IRCAM 1993.

[Mathews.69] **Mathews MV :** *The technology of Computer Music* – M.I.T. Press – 1969.

[Nigay 94] **Nigay L**. : *Conception et modélisation logicielle des systèmes interactifs : application aux interfaces multimodales* – Thèse à l'Université J.Fourrier – Grenoble, janvier 1994

[Norman.95] **Norman DA** : *The Psychopathology of Everyday Things* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.1 - Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann, San Fransisco, 1995. – extraits de l'ouvrage[Norman.88].

[OFTA.96] Nouvelles Interfaces Hommes Machines – Recueil de textes, coordonné par J Caelen – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Pierce.83] **Pierce J** : *Le son musical, musique, acoustique et informatique* – L'univers des sciences, pour la science, Belin, 1983.

[Poulain&Valot.96] **Poulain G, Valot C** : *Ergonomie des Interfaces Futures* – in *Nouvelles interfaces Homme-machine* – publié par l'Observatoire français des Techniques Avancées – série Arago n°18 - OFTA ed – Paris, 1996.

[Racot&al.99] **Racot G, Teruggi D, Favreau E** : *Evolution des outils, évolution des idées* in *Interfaces homme-machine et création musicale* – sous la direction de H Vinet et F Delalande – Hermès, 1999.

[Risset.96] : Recherche musicale "créative" - Assises de la Recherche. Culture & Recherche n°59 - Juillet 1996

[Roads.78] **Roads C** : *Automated Granular Synthesis of Sound* – CMJ vol 2/2 – MIT Press1978.

[Roads.91] **Roads C** : *Asynchronous Granular Synthesis* – in *Representation of Musical Signals* – G De Poli, A Piccialli and C Roads ed. - MIT Press - 1991.

[Roads&al.89] *The Music Machine* – edited by C Roads – MIT Press 1998

[Rossing.90] **Rossing TD** : *The Science of Sound* – 2nd edition - Addison Wesley Publishing Company - 1990.

[Schaeffer.52] **Schaeffer P** : *A la recherche d'une musique concrète* – Seuil - Paris 1952.

[Shneiderman.95] **Shneiderman B** : *A Taxonomy of Rule Base for the Selection of Interaction Styles* – in *Readings in Human Computer Interaction : Toward the Year 2000* – ch.6 pp. 401-411- Baecker R, Grudin J, Buxton WAS, Greenberg, S, ed – Moran Kaufmann - San Fransisco, 1995.

[Smith.93] **Smith III JO** : *Observations sur l'histoire de la synthèse numérique du son* – Cahiers de l'IRCAM n°2 : *la synthèse sonore* - ed IRCAM 1993.

[Taub.71] Taub AH : *Studies in Applied Mathematics* – Mathematical Association of America - 1971

[Vinet&al.99] *Interfaces homme-machine et création musicale* – Recueil de textes, sous la direction de Hugues Vinet et François Delalande – Hermès, 1999.

Table des illustrations

INTRODUCTION

Figure 1 : Organisation générale du document p.13

PARTIE I : LE PARADIGME DE LA SIMULATION MULTISENSORIELLE INTERACTIVE D'OBJETS PHYSIQUES

Figure 1 : la relation instrumentale..... p.18
Figure 2 : les quatre éléments de l'outil de création :Calculateur, formalisme moteur de simulation, périphériques et environnement..... p.19
Figure 3 : L'interaction simulation/transducteur nécessite deux canaux unidirectionnels de sens opposés..... p.22
Figure 4 : Les deux types de point de communication entre la simulation et l'extérieur : point M et point L. p.23
Figure 5 : Surface d'un modèle répondant au formalisme CORDIS-ANIMA..... p.24
Figure 6 : Règles de connexion entre modèles..... p.24
Figure 7 : forme externe des modules élémentaires <MAT> et <LIA>..... p.25
Figure 8 : communication TGR / simulation dans le cas d'une simulation 3D..... p.26
Figure 9 : T-simulation p.27
Figure 10 : représentation graphique symbolique (gauche) et figurative (droite) dans le modeleur du système ANIMA [Razafindrakoto.86]..... p.35
Figure 11 : objets simulés dans ANIMA p.35
Figure 12 : capture d'écran de MIMESIS..... p.37
Figure 13 : images générées après habillage dans MIMESIS p.37
Figure 14 : le Theremin Vox..... p.40
Figure 15 : Constituants des « instruments virtuels » d'après [Wanderley.99]..... p.48

PARTIE II : INTERACTION HOMME-MACHINE ET METHODOLOGIE DE CONCEPTION DE GENESIS

Figure 1 : l'interface GENESIS..... p.64
Figure 2 : facteurs en jeu pour la conception de l'interface GENESIS..... p.67
Figure 3 : limites absolues pour la perception visuelle d'après [Baecker & al.95e]..... p.68
Figure 4 : la loi de Fitts..... p.71
Figure 5 : le processeur humain de Card, Moran et Newell p.72
Figure 6 : les 7 étapes de l'action d'après [Norman.88]..... p.74
Figure 7 : les trois modes de la pensée dans le cas de l'interaction avec l'ordinateur – d'après [Kay.90] p.77
Figure 8 : reconnaissance de gestes d'après [Beaudouin.99]..... p.82
Figure 9 : reconnaissance de forme d'après [Baudel.96]..... p.82
Figure 10 : le principe du tool glass. [Bier&al.95]..... p.83
Figure 11 : le modèle Seeheim..... p.85
Figure 12 : le modèle Arch p.85
Figure 13 : le modèle MVC p.86
Figure 14 : agent PAC et hiérarchie d'agents PAC..... P.87
Figure 15 : modèles conceptuel d'après Norman, [Norman.88]..... p.92

Figure 16 : quelques conseils d'après [Norman.88] p.94
Figure 17 : processus et convention de versionnage de GENESIS p.110
Figure 18 : plusieurs catégories de méthodes pour l'évaluation d'après [Naël.96]... p.111
Figure 19: les boucles imbriquées de la méthodologie itérative adoptée pour la conception de GENESIS..... p.119

PARTIE III : LE PROCESSUS DE CREATION AVEC GENESIS ; ANALYSE DES TACHES

Figure 1 : méthodologie pour l'analyse des tâches..... p.124
Figure 2 : les deux temps de la création du son..... p.146
Figure 3 : tâches fondamentales de la phase de lutherie. P.148
Figure 4 : la simulation en mode lutherie. Usages et fonctionnalités. P.150
Figure 5 : démarche itérative en mode lutherie : la boucle modification – simulation – évaluation..... p.150
Figure 6 : catalogue, insertion de structures. P.151
Figure 7 : la chaîne instrumentale augmentée au sein des objets CORDIS-ANIMA..... P.152
Figure 8 : exploration paramétrique systématique..... p.156
Figure 9 : principes de l'analyse et de la génération..... p.157
Figure 10 : sorties possibles de la phase de lutherie. p.158
Figure 11 : récapitulatifs des tâches fondamentales impliquées dans la phase de Lutherie. ..... p .159
Figure 12 : boucles pour l'edition de la trace. p.162

PARTIE IV : CHOIX GENERAUX POUR L'ENVIRONNEMENT GENESIS

Figure 1 : changement de base de paramètres et algorithmes des modules CEL et REF topologiques..... p.175
Figure 2 : représentation de la courbe caractéristique du module FRO..... p.177
Figure 3 : un cas de spatialité complexe p.178
Figure 4 : deux représentations d'un objet topologique..... p.178
Figure 5 : modèle bidimensionnel pour une percussion élémentaire..... p.180
Figure 6 : déroulement d'une percussion d'un modèle 2D p.181
Figure 7 : algorithme de la partie Raideur Non Linéaire du LNL en pseudo code . p.188
Figure 8 : quelques exemples d'utilisation du module LNL. p.190
Figure 9 : modes de GENESIS et conditions de passage entre modes p.203
Figure 10 : différents types de données dans GENESIS..... p.205
Figure 11 : les différents formats de fichiers dans GENESIS. p.208
Figure 12 : lien causal entre les différents fichiers ; exportation..... p.209
Figure 13 : le processus de versionnage dans GENESIS..... p.211
Figure 14 : arborescence des comptes utilisateurs..... p.212
Figure 15 : organisation générale de l'établi. p.217
Figure 16 : catégories de fenêtres dans l'atelier de lutherie de GENESIS..... p.218

PARTIE V□ FONCTIONNALITES FONDAMENTALES DE L'ATELIER DE LUTHERIE GENESIS

Figure 1 : représentation. p.226
Figure 2 : cas particulier d'émergence..... p.228
Figure 3 : l'établi du luthier, espace de représentation..... p.230
Figure 4 : la représentation proposée par la version 0 de GENESIS..... p.230
Figure 5 : la représentation proposée par la version 1 de GENESIS..... p.231
Figure 6 : propositions pour la représentation des LIA orientés..... p.232
Figure 7 : représentation typologique des modules dans GENESIS 1.5 ..... p.233
Figure 8 : le même modèle à trois taux de zoom différents..... p.234
Figure 9 : trois représentations du même objet.
Gauche : représentation GENESIS
Centre : représentation fonctionnelle dans l'espace de simulation
Droite : représentation fonctionnelle. p.235
Figure 10 : aperçu général du modèle unique de la pièce Pico..TERA (Cadoz)..... p.236
Figure 11 : un modèle de trombone (Torvik)..... p.237
Figure 12 : un exemple de disposition évoquant l'objet modélisé..... p.237
Figure 13 : rappel des types de paramètres..... p.239
Figure 14 : répartition statistique typique des valeurs
d'un paramètre M, K ou Z dans un modèle CORDIS-ANIMA p.240
Figure 15 : trois modules MAS
dont le paramètre M appartient à trois classes différentes. p.241
Figure 16 : représentation des modules RES, FRO, REF et BUT
avec leurs paramètres. p.241
Figure 17 : représentation d'un module CEL avec ses paramètres K et Z. p.241
Figure 18 : la palette graphique de GENESIS p.247
Figure 19 : le menu des outils d'organisation du réseau p.247
Figure 20 : méta outil avec le modificateur ALT. p.248
Figure 21 : trois modules sélectionnés p.249
Figure 22 : arbre de décision pour la sélection par manipulation directe..... p.250
Figure 23 : quelques modules MAT désignés..... p.251
Figure 24 : arbre de décision pour l'ajout de modules LIA..... p.252
Figure 25 : aperçu (en haut à droite)..... p.254
Figure 26 : navigation par manipulation directe et raccourcis. p.255
Figure 27 : la fenêtre de résumé p.255
Figure 28 : fenêtre d'édition des paramètres de GENESIS 1.2..... p.257
Figure 29 : flux d'interface avec la fenêtre des paramètres de GENESIS 1.2..... p.257
Figure 30 : principe d'homogénéité pour les paramètres p.258
Figure 31 : fenêtre d'édition des paramètres de GENESIS 1.3 ..... p.259
Figure 32 : flux d'interface avec la fenêtre des paramètres de GENESIS 1.3..... p.259
Figure 33 : flux d'interface pour la fenêtre des paramètres
et la fenêtre des conditions initiales dans GENESIS 1.5 ..... p.261
Figure 34 : validation et annulation..... p.261
Figure 35 : fenêtre d'édition des paramètres de GENESIS 1.5 ..... p.262

Figure 36 : fenêtre d'édition des conditions initiales de GENESIS 1.5 ..... p.263
Figure 37 : flux d'interface pour la fenêtre des conditions initiale p.264
Figure 38 : prévisualisation et édition de l'amplitude des caractéristique des LNL dans la fenêtre des paramètres de GENESIS 1.5 ..... p.266
Figure 39 : la fenêtre bloquante pour l'édition des courbes des LNL p.267
Figure 40 : flux d'interface pour l'édition des LNL p.267

PARTIE VII : FONCTIONNALITES EVOLUEES DU MODE LUTHERIE

Figure 1 : définition d'un ensemble p.281
Figure 2 : identification des tâches relatives aux Ensembles p.282
Figure 3 : les deux aspects de la fenêtre des Ensembles de GENESIS 1.5 ..... p.283
Figure 4 : surface structurelle d'une capsule p.284
Figure 5 : capsules contenant une corde CORDIS-ANIMA p.286
Figure 6 : complémentarité des Ensembles et Capsules p.287
Figure 7 : la chambre modale p.291
Figure 8 : évolution de TAUX en fonction du nombre d'itération et du temps p.299
Figure 9 : communication entre GENESIS et le programme AnModTop p.300
Figure 10 : la fenêtre simple d'analyse p.301
Figure 11 : la fenêtre de détail de l'analyse modale p.302
Figure 12 : l'outil de génération de structures régulières p.303
Figure 13 : quelques structures générées. Anneau, membranes, spyrale p.304
Figure 14 : la génération est suivie d'une adaptation p.304
Figure 15 : complexification progressive des propriétés physiques d'un objet par substitution de modules p.305
Figure 16 : substitutions p.305
Figure 17 : fenêtre de substitution. p.306
Figure 18 : la fenêtre de multiplication des paramètres de GENESIS 1.5. ..... p.307

PARTIE VIII : SIMULATION ET PHENOMENES

Figure 1 : un objet en mouvement (GENESIS 1.2) p.322
Figure 2 : deux représentations possibles des <LIA> p.323
Figure 3 - rappel : la boucle modification / simulation / évaluation p.324
Figure 4 : répartition des enjeux suivant les modalités d'observation des simulations p.324
Figure 5 : une première option : visualtion de l'objet en mouvement dans l'établi p.327
Figure 6 : modularité de la fenêtre de simulation p.329
Figure 7 : la fonctionnalité Ecouter dans la fenêtre de simulation (GENESIS 1.5) p.330
Figure 8 : deux options pour l'organisation de l'axe de simulation p.331

Figure 9 : la fonctionnalité Regarder dans la fenêtre de simulation (GENESIS 1.5) p.333
Figure 10 : répartition des procédés de communication suivant l'état du simulateur p.336
Figure 11 : communication entre GENESIS et les Simulateurs p.337
Figure 12 : affichage de l'objet en mouvement p.339

ANNEXE A : MODELISATION PHYSIQUE, SON, MUSIQUE

Figure 1 : définitions des techniques de synthèse Modèle, instrument, paramètres, flux dynamique p.356
Figure 2: tableau résumé des critère d'évaluation comparée p.375
Figure 3 : l'algorithme Karplus-Strong p.377
Figure 4 : l'approche source filtre en interaction p.379

ANNEXE B : ETUDE D'UN OBJET CORDIS-ANIMA BIDIMENSIONNEL

Figure 1: building a model p.386
Figure 2: 3D-appearance models p.387
Figure 3: a 0D-appearance model p.387
Figure 4: a 2D-qppearance model p.387
Figure 5: the 2D string-like model p.388
Figure 6: sonogram of a model that shows off the first non-linearity p.388
Figure 7: picture of a model that shows off the first non-linearity during simulation p.389
Figure 8: sonogram of a model that shows off the second non-linearity p.389
Figure 9: picture of a model that shows off the second non-linearity during simulation p.389
Figure 10: simplifications hypothesis p.389
Figure 11: the 1D-simulation space equivalent model p.390
Figure 12: the new basic interaction p.390
Figure 13: control of the non-linear parameter p.391

Total : 146 figures.
Les figures de l'annexe 3, non listées ici, sont de Claude Cadoz.

Table des matières

AVANT-PROPOS 5

SOMMAIRE 7

INTRODUCTION 9

PARTIE I : LE PARADIGME DE LA SIMULATION MULTISENSORIELLE INTERACTIVE D'OBJETS PHYSIQUES 15

Chapitre 1 Le contexte de nos travaux 18

1.1 - La relation instrumentale comme situation de référence 18
1.2 - Périphériques gestuels - retour d'effort 20
1.2.1 - Une Typologie du geste instrumental 20
1.2.2 - La famille des TGR© 21
1.3 - CORDIS-ANIMA 22
1.3.1 - Interface transducteur - simulation : point de communication 22
1.3.2 - Principes générateurs du formalisme 23
1.3.3 - Réseaux d'interaction par les points M et L - règles de connexion 23
1.3.4 - Elément matériel <MAT> et élément de liaison <LIA> 25
1.3.5 - Dimension de l'espace de simulation et spatialité des objets CORDIS-ANIMA 25
1.3.6 - Principes généraux des algorithmes CORDIS-ANIMA 26
1.3.7 - Principaux modules 28
1.3.8 - Implémentation de CORDIS-ANIMA□ moteurs de simulation CORDIS-Off et TELLURIS 33
1.4 - Environnements pour la création 34
1.4.1 - Environnement pour la création graphique 34
1.4.2 - Environnement pour la création musicale 37
1.5 - Conclusion: Ici apparaît le paradigme de la simulation Multisensorielle Interactive d'Objet Physique 34

Chapitre 2 Nouveaux outils, nouvelles démarches 39

2.1 - Les instruments électroacoustiques 40
2.2 - L'enregistrement sonore 42
2.3 - La synthèse, l'analyse, la psychoacoustique 44
2.4 - La Composition Assistée par Ordinateur et les processus d'engendrement 46
2.5 - Les interfaces gestuelles, les systèmes temps réel 48
2.6 - La simulation interactive et multisensorielle d'objets physiques 51
2.6.1 - Une étape du mouvement vers une meilleure sensibilité des outils 51
2.6.2 - L'objectivation du geste musical□ la trace gestuelle□ composition du geste 53
2.6.3 - CORDIS-ANIMA, un système pour la modélisation physique 54
2.6.4 - Une musique des corps sonores□ 56
2.6.5 - Continuité de la forme au matériau et unification des processus de génération : le modèle physique, métaphore pour la composition 57

Conclusion□penser physique 59

PARTIE II□□INTERACTION HOMME-MACHINE ET MÉTHODOLOGIE DE CONCEPTION DE GENESIS 61

Chapitre 3 Interface, Homme, Ordinateur 64

3.1 - Interface, système interactif 64
3.1.1 - Définition 64
3.1.2 - Vertus premières 65
3.1.3 - Conception 66

3.2 - L'homme dans son environnement 67
3.2.1 - Psychophysique et interaction homme-ordinateur : données pratiques sur l'homme percevant et agissant 67
3.2.2 - L'approche « computationnelle » et les modèles de l'homme face à ses outils 71
3.2.3 - Action, perception, représentation mentale : l'approche écologique de Gibson 75
3.2.4 - Action, représentation, symboles : les modes de la pensée selon Bruner 76
3.3 - L'ordinateur « communiquant » 78
3.3.1 - Interfaces graphiques WIMP 78
3.3.2 - Styles d'interaction courants : *Widgets*, manipulation directe, interaction par langage 79
3.3.3 - Quelques innovations et tendances récentes en matière d'interaction homme-ordinateur 81
3.3.4 - Architecture logicielle des systèmes interactifs : modèles 84

Chapitre 4 Guides et Principes ergonomiques 89

4.1 - Les principes inspirés de Hansen. 89
4.1.1 - Connaître l'utilisateur et ce qu'il veut faire 90
4.1.2 - Minimiser la mémorisation 90
4.1.3 - Optimiser les opérations 90
4.1.4 - Gérer correctement les erreurs 91
4.2 - Modèle mental, *affordance*, contrainte, *mapping*, visibilité et retour : l'analyse de Norman 91
4.2.1 - Modèle mental 91
4.2.2 - Affordance 92
4.2.3 - Contraintes 93
4.2.4 - Visibilité et retour (*feedback*) 93
4.2.5 - Projection (*Mapping*) 94
4.2.6 - Les conseils de Norman 94
4.3 - Métaphores 94
4.4 - Economie, simplicité et souci du détail 96
4.5 - Cohérence et conformité 97
4.6 - Quelques conseils pour une bonne communication visuelle 98
4.6.1 - Les conseils de Marcus 98
4.6.2 - De l'utilisation de la couleur dans les interfaces graphiques 99
4.7 - Objet d'intérêt et manipulation directe : « Moins d'interface pour plus d'interaction » 100
4.7.1 - Lorsque sont trahis les principes de la manipulation directe… 100
4.7.2 - La manipulation directe dans les modeleurs/simulateurs 101
4.8 - Théorie et principes *vs* pragmatisme et compromis 102

Chapitre 5 Méthode pour la réalisation de GENESIS 104

5.1 - Première étape : Analyse des tâches 104
5.1.1 - Un processus récursif 104
5.1.2 - Les processus de création 105
5.1.3 - Scénarios 106
5.2 - Deuxième étape : Propositions, décision 107
5.2.1 - Recherche et genèse des idées 107
5.2.2 - Retour au cahier des charges, évaluation, décision 108
5.2.3 - Un cas particulier : l'ergonomie « mise à plat » 108
5.3 - Troisièmé étape : Implémentation : génie logiciel 109
5.3.1 - Spécifications techniques 109
5.3.2 - Développement parallèle et intégration 109
5.3.3 - Le versionnage de GENESIS 109
5.4 - Evaluation et tests 111
5.4.1 - Méthodes analytiques 111
5.4.2 - Tests par les développeurs et « *Essayez de tout casser !* » 112
5.4.3 - Confrontation aux utilisateurs 113
5.4.4 - Conclusion sur l'évaluation 116
5.5 - Démarche incrémentale et itérative : les boucles imbriquées 117
5.5.1 - Démarche incrémentale 117
5.5.2 - Démarche itérative, boucles imbriquées 117

Conclusion 120

PARTIE III : PROCESSUS DE CRÉATION AVEC GENESIS ANALYSE DES TÂCHES 123

Chapitre 6 Un parcours simple pour la création 126

6.1 - L'objet le plus simple 126
6.2 - Structures vibrantes 129
6.3 - Com-position de structures vibrantes.
Chevalet, environnement local, environnement global. 132
6.4 - Excitateur, modificateur, déclencheur 135
6.5 - Instrumentiste, chef, déclencheurs 137
6.6 - Fin de la phase de conception 139
6.7 - Jeu et interaction multisensorielle 140
6.8 - L'objet "trace" et l'incrémentation du jeu 140
Edition de la trace 141

Chapitre 7 Considérations générales sur la création avec CORDIS-ANIMA 143

7.1 - GENESIS, un environnement complet 143
7.2 - Situation temps-réel et temps-différé : la simulation d'objets physiques 144
7.3 - Composantes structurelle et temporelle du son ; les deux temps de la création 145

Chapitre 8 Phase de lutherie : composition d'objets physiques 147

8.1 - Tâches élémentaires 148
8.2 - La simulation en phase de lutherie 149
8.3 - Stratégies itératives et incrémentales 150
8.4 - Conception hiérarchique 151
8.5 - Les notions d'échelle 153
8.6 - La macro-modularité dans le processus de lutherie 154
8.7 - Détermination des paramètres : exploration paramétrique 155
8.8 - Relations et changement de base de paramètres 156
8.9 - Expertise, analyse et génération : vers un contrôle explicite des propriétés 156
8.10 - Commentaires, légendes 157
8.11 - Différentes sorties de la phase de lutherie 158

Chapitre 9 Phase de jeu et autres aspects de la création avec GENESIS 160

9.1 - Jouer avec l'instrument : interaction multisensorielle via le TGR© 161
9.2 - D'autres flux de données 161
9.3 - Enregistrement et relecture de traces gestuelles 162
9.4 - Edition du signal trace : la boucle d'édition de la trace 162
9.5 - D'autres horizons 164

Résumé et Conclusion 165

PARTIE IV : CHOIX GÉNÉRAUX POUR L'ENVIRONNEMENT GENESIS..... 167

Chapitre 10 La version «□GENESIS□ de CORDIS-ANIMA.....170

10.1 - Restriction au temps différé.....170
10.2 - Fréquence de simulation.....171
10.2.1 - Une fréquence unique□44100 Hz.....171
10.2.2 - Remarques.....172
10.3 - Base de paramètres dans GENESIS.....172
10.3.1 - Paramètres algorithmiques, paramètres CORDIS, paramètres naturels.....172
10.3.2 - Choix retenus pour GENESIS.....176
10.3.3 - Remarques.....176
10.4 - La spatialité des objets CORDIS-ANIMA et le choix de la version topologique.....177
10.4.1 - Espace des mouvements, géométrie, espace de simulation□ les trois caractères de la spatialité.....177
10.4.2 - Quelques enjeux de la spatialité.....179
10.4.3 - Expériences□décisions.....182
10.5 - La version topologique de CORDIS-ANIMA utilisée dans GENESIS.....183
10.5.1 - A propos des modules linéaires élémentaires et intégrés.....183
10.5.2 - A propos des modules non-linéaires.....184
10.5.3 - A propos des modules d'écoute et de mesure.....192
10.5.4 - A propos des modules d'entrée (et d'interaction).....193
10.5.5 - A propos des modules fonctionnels de modification dynamique paramétrique et structurelle.....193
10.5.6 - Panorama résumé des modules CORDIS-ANIMA implémentés dans GENESIS.....194

Chapitre 11 Organisation générale de GENESIS.....201

11.1 - Les différents modes et les fonctionnalités de simulation.....201
11.1.1 - lutherie, jeu, composition, gestion□les modes d'une version idéale de GENESIS.....201
11.1.2 - La simulation, un paradigme central.....203
11.1.3 - L'implémentation réalisée□restriction au mode lutherie.....204
11.2 - Fichiers et comptes utilisateurs.....205
11.2.1 - Les différents types de données□les différents formats de fichiers.....205
11.2.2 - Objet, analyses et phénomènes : causalité et versionnage.....208
11.2.3 - Arborescence des comptes utilisateurs.....211
11.3 - L'interface du mode Lutherie.....213
11.3.1 - GENESIS, une interface WIMP.....213
11.3.2 - L'atelier de lutherie GENESIS.....215

Résume et Conclusion.....220

PARTIE V : FONCTIONNALITÉS FONDAMENTALES DE L'ATELIER DE LUTHERIE GENESIS 223

Chapitre 12 Représentation des objets CORDIS-ANIMA 226

12.1 - Enjeux et problèmes 226
12.1.1 - L'objet CORDIS-ANIMA est un réseau de modules paramétrés 226
12.1.2 - Echelle et inhomogénéité216 227
12.1.3 - Des propriétés émergentes… 227
12.1.4 - Soutenir « impression de réalité »
représentation réaliste *versus* représentation schématique 228
12.1.5 - Surcharge de la représentation 229
12.1.6 - Esthétique. Simplicité, efficacité 229
12.1.7 - Des contraintes techniques 229
12.2 - Pistes 230
12.2.1 - L'établi : espace de représentation 230
12.2.2 - Représentation des types des modules et du réseau. 230
12.2.3 - Taille de l'établi, zoom, navigation 234
12.2.4 - De l'utilité de l'espace de représentation 235
12.2.5 - Représentation des paramètres : classes de paramètres 239
12.3 - Résumé des choix opérés 243

Chapitre 13 Manipulation directe du réseau 244

13.1 - Enjeux et difficultés 244
13.1.1 - Manipulation directe et modèle mental 244
13.1.2 - Les enjeux de la manipulation directe dans GENESIS 244
13.1.3 - De nombreuses fonctionnalités, peu de moyens 245
13.2 - Pistes et réalisations 246
13.2.1 - Outil et palette 246
13.2.2 - La sélection, préalable à toute action 248
13.2.3 - Ajout de modules 251
13.2.4 - Déconnexion, re-connexion des <LIA> 252
13.2.5 - Suppression de modules 252
13.2.6 - Organisation de l'objet sur l'établi 253
13.2.7 - Navigation dans l'établi : outils de navigation 253
13.2.8 - Inspection : la fenêtre de résumé 255

Chapitre 14 Edition des paramètres et conditions initiales 256

14.1 - Etude de cas : Evolution de la fenêtre d'édition des paramètres 256
14.1.1 - Edition des paramètres dans GENESIS 1.2 256
14.1.2 - GENESIS 1.3 259
14.2 - Paramètres et conditions initiales dans GENESIS 1.5 260
14.2.1 - Deux types de données, deux ergonomies 260
14.2.2 - Propriétés communes aux deux fenêtres 260
14.2.3 - La fenêtre des paramètres 262
14.2.4 - Fenêtre des conditions initiales 262
14.3 - Edition des liaisons non linéaires LNL 265
14.3.1 - Les LNL dans la fenêtre des paramètres :
visualisation des caractéristiques et modification d'amplitude 266
14.3.2 - La fenêtre d'édition profonde 266

Chapitre 15 Perspectives pour les fonctionnalités fondamentales 268

15.1 - Echelles et valeurs par défaut 268
15.2 - Extension de la manipulation directe et de la représentation 270
15.3 - Trans-représentation et multimodalité 271

Résume et Conclusion 273

PARTIE VI : FONCTIONNALITÉS ÉVOLUÉES DU MODE LUTHERIE..... 275

Chapitre 16 Macro-modularité : Ensembles et capsules.....278

16.1 - Une analyse du problème.....278
16.1.1 - Enjeux.....278
16.1.2 - Typologie des ensembles de modules dans GENESIS.....280
16.1.3 - Deux notions, deux fonctionnalités.....281
16.2 - Ensembles.....281
16.2.1 - Définition et utilisations.....281
16.2.2 - Identification des tâches relatives aux Ensembles.....281
16.2.3 - Ergonomie.....282
16.3 - Capsules.....284
16.3.1 - Définition.....284
16.3.2 - Implémentation dans GENESIS 1.5.....285
16.4 - Conclusion sur les fonctionnalités macro-modulaires.....287

Chapitre 17 Analyse modale et accordage.....288

17.1 - Approche théorique de l'analyse modale.....288
17.1.1 - Formulation matricielle des algorithmes.....288
17.1.2 - Cas des objets à viscosité compatible.....290
17.1.3 - La chambre modale.....291
17.1.4 - Enseignements de la représentation modale.....292
17.1.5 - Utilisation du système modal : accordage.....294
17.1.6 - Cas des objets à viscosité non compatible.....296
17.2 - L'analyse modale dans GENESIS.....298
17.2.1 - L'utilitaire AnModTop.....298
17.2.2 - Ergonomie.....300

Chapitre 18 Outils complémentaires d'édition.....303

18.1 - Génération de structures régulières.....303
18.2 - Substitution de modules.....305
18.3 - Multiplication des paramètres et maîtrise des échelles.....306

Chapitre 19 Perspective pour les fonctionnalites evoluees.....308

19.1 - Edition évoluée des paramètres : relations et meta-paramètres.....308
19.1.1 - Motivation.....308
19.1.2 - Relation et meta-paramètres.....309
19.1.3 - Analyse des difficultés.....311
19.2 - Evolution des fonctionnalités de macro-modularité.....313
19.2.1 - Evolution des Ensembles.....313
19.2.2 - Evolutions des capsules : surface paramétrique.....314
19.3 - Autres fonctionnalités.....314
19.3.1 - Substitution évoluée.....314
19.3.2 - Génération évoluée.....315
19.3.3 - Exploration paramétrique systématique.....315

Résume et Conclusion.....316

PARTIE VII : SIMULATION ET PHÉNOMÈNES 319

Chapitre 20 La fenêtre de simulation 322

20.1 - La simulation dans GENESIS 1.2 à 1.4 322
20.2 - Analyse des enjeux et des tâches 324
20.2.1 - Trois enjeux, trois modalités d'observation 324
20.2.2 - Tâche pour la fonctionnalité Ecouter 325
20.2.3 - Tâches pour la fonctionnalité Regarder 326
20.2.4 - Un quatrième enjeu pour l'ergonomie 327
20.3 - Fonctionnalités et ergonomie de la fenêtre de simulation 327
20.3.1 - La fenêtre de simulation 327
20.3.2 - Conservation du lien causal 329
20.3.3 - Ecouter 330
20.3.4 - Regarder 331
20.4 - Notes sur l'implémentation 335

Chapitre 21 Perspectives pour les fonctionnalités de simulation 338

21.1 - Finalisation 338
21.1.1 - Evaluation 338
21.1.2 - Incrémentation 338
21.1.3 - Optimisation 339
21.2 - Métrologie 340
21.3 - Vers une véritable multisensorialité : connexion au moteur temps-réel TELLURIS 341

CONCLUSIONS ET PERSPECTIVES 345

ANNEXES 353

Annexe A Modélisation physique, Son, Musique **354**
1. Le *modèle physique*, une notion porteuse d'espérances 355
1.1. Quelques définitions 355
1.2. Modèles de signaux / modèles physiques 356
1.3. Le modèle physique en question 358
1.4. Son d'instruments traditionnels ; mimétisme 359
1.5. Les modèles physiques, "générateurs de réalité musicale" 360
1.6. Modèles physiques, causalité acoustique et objectif de "plausibilité" 361
1.7. Modèle physique, comportement dynamique, jeu ; accident et phrasé 362
2. Des critères pour une évaluation des techniques de synthèse et de modélisation physique pour la musique 364
2.1. Des critères pour une évaluation des techniques de synthèse 365
2.2. Evaluation globale des modèles physiques en regard des critères 371
2.3. Des critères spécifiques aux modèles physiques 372
3. Principales techniques de modélisation physique pour la musique 376
3.1. Analyse numérique, éléments finis, différences finies... Les outils de l'acousticien 376
3.2. L' "approche classique" des travaux de Ruiz et Hiller 376
3.3. Les guides d'ondes 377
3.4. L'approche source / filtre en interaction, techniques numériques non linéaires 379
3.5. La synthèse modale 380
3.6. L'approche particulaire : CORDIS-ANIMA 383

Annexe B Etude d'un objet CORDIS-ANIMA bidimensionnel **385**

Annexe C « Simuler pour connaître Connaître pour simuler » (Claude Cadoz) **393**

Annexe D Quelques échanges avec les utilisateurs **433**

Annexe E Plages couleur **443**

RÉFÉRENCES BIBLIOGRAPHIQUES 449

TABLE DES ILLUSTRATIONS 463